



THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par l'Université Toulouse III Paul Sabatier
Discipline ou spécialité : Informatique

Présentée et soutenue par *Mohamed EL Amine BOUABID*
Le 07 Décembre 2012

Titre :

*De la Conception à l'Exploitation des Travaux Pratiques en ligne:
Application d'une Approche Générique à l'Enseignement de l'Informatique*

JURY

Jean-Marc LABAT, Professeur à l'Université Pierre et Marie Curie, Paris, Président
Christophe CHOQUET, Professeur à l'Université du Maine, Laval, Rapporteur
Sébastien GEORGE, MCF (HDR) à l'INSA de Lyon
DR, Ferhat KHENAK, Responsable du Pôle TICE, CERIST, Alger, Algérie
Julien BROISIN, MCF, Université Toulouse III Paul Sabatier, Encadrant

Ecole doctorale : *Mathématiques, Informatique, Télécommunications (MITT)*
Unité de recherche : *Institut de Recherche en Informatique de Toulouse CNRS - UMR - 5505*
Directeur(s) de Thèse : *Philippe VIDAL, Professeur à l'Université Toulouse III, Paul Sabatier*
Rapporteurs : *Noms des rapporteurs (s'ils ne font pas partie des membre du jury)*

Résumé

La mise en ligne des travaux pratiques est une problématique de recherche étudiée depuis plus d'une décennie en raison des enjeux pédagogiques et économiques qu'elle engendre. En effet, il s'agit d'une catégorie d'enseignements indispensables aux disciplines scientifiques, technologiques et mathématiques et d'ingénierie visant à atteindre des objectifs d'apprentissage particuliers liés à l'acquisition de compétences professionnelles et scientifiques, mais aussi au travail de groupe et de communication. Des recherches dans le domaine des télé-TPs ont permis de mettre en évidence des problèmes d'ordre pédagogiques et techniques qui entravent le développement et l'acceptation des télé-TPs par les acteurs de la formation, dont nos travaux tentent d'apporter des réponses.

En particulier, cette thèse traite les problématiques liées (1) aux fondements théoriques de l'apprentissage en ligne dans le cadre de télé-TPs, (2) aux avancées tant techniques que pédagogiques des Environnements Informatiques pour l'Apprentissage Humain (EIAH), et (3) aux avancées techniques et technologiques pour la mise en ligne de dispositifs de laboratoires distribués et mutualisés. En d'autres termes, il s'agit d'atteindre l'efficacité pédagogique des télé-TPs tout en les intégrant aux EIAHs utilisés pour les autres types d'activités d'apprentissage.

Nos propositions s'articulent autour de deux éléments clés : (1) une architecture structurée en couches assurant une communication transparente entre les contextes d'apprentissage (EIAH) et d'expérimentation (laboratoires en ligne), et (2) un méta-modèle qui, d'une part, décrit de façon formelle les éléments inhérents à l'activité de télé-TP (laboratoires, dispositifs et expériences) ainsi que les activités réalisées par les utilisateurs sur ces entités, et d'autre part dirige le contrôle et la supervision des expériences et laboratoires. D'un point de vue technique, nos contributions concernent la proposition d'un méta-modèle standard pour le pilotage et la gestion du cycle de vie des expériences qui facilite la représentation homogène des dispositifs hétérogènes de laboratoires tout en favorisant le déploiement et l'exploitation d'expériences distribuées sur plusieurs sites, ainsi que l'introduction d'interfaces de contrôle et de supervision uniformes masquant les détails spécifiques des laboratoires distants aux

applications de haut niveau telles que les EIAHs. D'un point de vue pédagogique, nous introduisons l'objet pédagogique « Expérience » afin d'encourager la pratique, le partage et la réutilisation des télé-TPs dans les cursus en ligne, et supportons différentes fonctions pédagogiques telles que l'apprentissage collaboratif et le tutorat (synchrone et asynchrone) à travers un ensemble d'IHMs adaptées dont le développement est rendu aisé par la décomposition du système global en différentes couches.

Nos propositions ont été expérimentées dans le cadre de télé-TPs relatifs à notre domaine d'enseignement, l'informatique. Les différents méta-modèles génériques ont été spécialisés pour décrire et superviser l'environnement d'expérimentation ainsi que les expériences qui y prennent place, mais également afin de modéliser, stocker et exploiter les traces des interactions entre les utilisateurs et ces expériences. Enfin, des composants spécifiques assurent le lien entre les modèles abstraits et les ressources de bas niveau, alors que des outils et IHMs de haut niveau supportent différentes fonctions pédagogiques :

- La conception pédagogique est étendue à la conception des expériences en ligne à travers une IHM intuitive proposée aux enseignants, qui leur permet de concevoir et de tester des expériences avec un niveau de complexité arbitraire.
- L'apprentissage en ligne est supporté par une interface Homme-Machine réaliste qui assure à la fois la réalisation d'opérations sur les expériences distantes, une navigation efficace parmi les composants d'une expérience en ligne, mais également l'apprentissage collaboratif.
- Le tutorat en ligne est supporté par une interface qui étend l'IHM d'apprentissage avec des privilèges élargis et des outils supplémentaires permettant aux tuteurs distants de détecter les situations de blocage, de venir en aide aux apprenants en difficulté, et enfin d'évaluer les travaux de ces apprenants.

Abstract

Bringing online practical learning activities is a research problem studied for over a decade due to educational and economic challenges it engenders. Indeed, it is a category of courses required for scientific, technological, engineering and mathematics disciplines to achieve specific learning objectives related to the acquisition of professional and scientific skills, but also teamwork and communication skills. Research in the field of remote lab-works helped to highlight pedagogical and technical problems hindering the development and acceptance of remote lab-works by the human actors, which our work attempt to provide answers.

In particular, this thesis addresses issues related (1) to the learning theoretical foundations in the context of remote lab-works; (2) Advances both technical and pedagogical of the Technology Enhanced Learning (TEL), and (3) Technical and Technological advances in online distributed and shared laboratories devices. In other words, it is to achieve the pedagogical effectiveness of remote lab-works while integrating them in LMS (Learning Management Systems) used for the other types of learning activities.

Our proposals center around two key elements: (1) A structured layered architecture ensuring seamless communication between learning context (LMS) and experimental context (online labs), and (2) A meta-model that, on one hand, describes formally the elements inherent to the lab-work activity (labs, experiments and devices) and the activities performed by users on these entities, and on the other hand, drives the control and supervision of laboratories and experiments.

From a technical standpoint, our contributions concern the proposal of a meta-model standard for controlling and managing the lifecycle of experiments that facilitates uniform representation of heterogeneous laboratories' devices while promoting the deployment and the operating of distributed experiments across multiple sites, as well as the introduction of uniform interfaces for control and monitoring, hiding the specific details of remote laboratories to high-level applications such as LMSs.

From an educational point of view, we introduce the learning object "Experiment" to encourage the practice, sharing and the reuse of remote lab-works in the online curriculum, and support the different educational functions such as collaborative learning and tutoring

(synchronous and asynchronous) through a set of tailored GUIs whose development is made easy by the decomposition of the overall system into different layers.

Our proposals have been tested in the context of remote lab-works on our field of education: The Computer Science. The various generic meta- models were specialized to describe and monitor the environment of experimentation and the experiments that take place within them, but also to model, store and use the traces of interactions between user and these experiments. Finally, specific components provide the link between abstract models and concrete low-level resources, whereas high-level HCI and tools support the different pedagogical functions:

- The instructional design is extended to the design of online experiments through an intuitive GUI offered to teachers, allowing them to design and test experiments with an arbitrary level of complexity.
- Online learning is supported by a realistic interface that ensures both performing operations on remote experiments; navigate efficiently among the components of an online experience, but also collaborative learning.
- The online tutoring is supported by an interface that extends the learning HCI with extended privileges and additional tools enabling remote tutors detect deadlocks, to help struggling learners, and finally to evaluate the work of these learners.

Dédicaces

*Je dédie ce modeste travail à celui qui m'a entouré, guidé et soutenu depuis ma naissance,
A la mémoire de mon défunt père **Mohammed El Hadi***

Remerciements

Je tiens en premier lieu à remercier mes directeurs de thèse, Pr. Philippe Vidal et Dr. Julien Broisin d'avoir accepté de diriger ce travail, de m'avoir appris la rigueur scientifique, pour leur suivi minutieux, leurs conseils, leur sympathie, leur soutien, leur patience, leur disponibilité et les facilités qu'ils m'ont toujours accordés pour la poursuite de cette thèse.

Je tiens ensuite à exprimer mes vifs remerciements et gratitude au Dr. Ferhat Khenak qui m'a encouragé à m'engager dans cette thèse, je le remercie pour son engagement sans faille à mes côtés pour l'aboutissement de ce travail, pour ses contributions scientifiques, pour son soutien moral et les innombrables services qu'il ma rendu.

Je tiens à remercier Dr. Christophe Choquet et Dr. Sébastien George, qui ont accepté d'être les rapporteurs de ce travail

Je remercie Pr. Jean Marc Labat qui a accepté de présider le jury de soutenance de cette thèse.

Mes remerciements vont à mes amis de toujours, en particulier ceux qui m'ont apporté l'aide qu'il faut quand il le faut : Issam, Samir, Si Youssef, Fayçal, Farid et Abdelkader.

Mes remerciements vont aussi aux personnels de l'IRIT/SIERA pour leur support tout au long de ce travail en particulier Mme Martine Labruyere, Pr. Michelle Sibilla pour son aide scientifique pertinente et les membres de notre équipe : Valentin, Triomphe et Olivier.

Je tiens à remercier mes collègues du CERIST pour leur soutien et leur sympathie, en particulier Madame Elmaouhab, mais également à toute personne ayant contribué, de près ou de loin, à la réussite de ce travail.

Enfin, ma profonde gratitude et reconnaissance vont à mon défunt Père et à ma Mère, mon épouse Louiza, mes enfants Sara et Abdelmalek, mes frères Lyes et Hamza et ma sœur Ouassila qui m'ont tous soutenu, chacun à sa façon durant les longues années de mes études. Je n'oublie pas d'adresser mes remerciements sincères à mes beaux parents, mes belles sœurs et beaux frères.

Table des matières

Résumé.....	1
Abstract	3
Dédicaces	5
Remerciements.....	6
Table des matières.....	7
Liste des tableaux.....	16
Liste des figures	17
Introduction	20
Chapitre 1. Introduction et problématiques abordées	21
1.1. Cadre scientifique	21
1.2. Contexte	22
1.3. Problématiques	24
1.3.1. Efficacité pédagogique.....	25
1.3.2. Intégration dans les EIAHs	26
1.3.3. Support des fonctions pédagogiques.....	26
1.3.4. L’IHM et la télé-opération	27
1.3.5. Mutualisation des ressources des laboratoires	27
1.3.6. Standardisation des expériences.....	28
1.4. Organisation de ce manuscrit	28
Partie I : Etat de l’art	31
Chapitre 2. Les travaux pratiques et les travaux pratiques à distance pour l’apprentissage de la science	32
2.1. Introduction	32
2.2. Caractérisation des travaux pratiques et télé-TPs	33
2.2.1. Définitions des TPs	33
2.2.2. Définition des laboratoires	35
2.2.3. Définition des expériences	36
2.3. Dimensions d’analyse de l’efficacité pédagogique	39

2.3.1.	Les objectifs pédagogiques	40
2.3.2.	La conception pédagogique.....	43
2.3.3.	Les activités d'apprentissage.....	47
2.3.4.	La présence.....	48
2.3.5.	La fidélité de la télé-instrumentation	49
2.3.6.	L'apprentissage collaboratif	50
2.3.7.	Le tutorat	52
2.3.8.	L' <i>Awareness</i>	53
2.4.	Intégration des télé-TPs en EIAH	56
2.4.1.	Introduction aux EIAH.....	56
2.4.2.	Les standards pédagogiques	58
2.4.3.	Positionnement des télé-TPs	58
2.5.	Synthèse	62
Chapitre 3.	Etat de l'art des télé-TPs	65
3.1.	Le modèle d'analyse	66
3.2.	Le projet PEARL	67
3.2.1.	Caractéristiques générales	67
3.2.2.	Caractéristiques du laboratoire distant	69
3.2.3.	Caractéristiques de l'EIAH	69
3.2.4.	Support à la conception pédagogique.....	70
3.2.5.	Support à l'apprentissage collaboratif et au tutorat en ligne.....	70
3.2.6.	Synthèse du projet PEARL	71
3.3.	Le projet Lab@Future	71
3.3.1.	Caractéristiques générales	71
3.3.2.	Caractéristiques du laboratoire distant	72
3.3.3.	Caractéristiques de l'EIAH	73
3.3.4.	Support à la conception pédagogique.....	73
3.3.5.	Support à l'apprentissage collaboratif en ligne	74
3.3.6.	Support au tutorat en ligne	74
3.3.7.	Synthèse	74
3.4.	Le projet FORMID	75
3.4.1.	Caractéristiques générales	75

3.4.2.	Caractéristiques du laboratoire distant	75
3.4.3.	Caractéristiques de l'EIAH	76
3.4.4.	Support à la conception pédagogique.....	76
3.4.5.	Support à l'apprentissage collaboratif et au tutorat en ligne	77
3.4.6.	Synthèse	78
3.5.	Le projet ICTT@LAB	78
3.5.1.	Caractéristiques générales	78
3.5.2.	Caractéristiques du laboratoire distant	79
3.5.3.	Caractéristiques de l'EIAH	80
3.5.4.	Support à la conception pédagogique.....	80
3.5.5.	Support à l'apprentissage collaboratif et au tutorat en ligne	80
3.5.6.	Synthèse	81
3.6.	Le projet eINST	81
3.6.1.	Caractéristiques générales	81
3.6.2.	Caractéristiques du laboratoire distant	82
3.6.3.	Caractéristiques de l'EIAH	83
3.6.4.	Support à la conception pédagogique.....	83
3.6.5.	Support à l'apprentissage collaboratif en ligne	83
3.6.6.	Support au tutorat en ligne	84
3.6.7.	Synthèse	84
3.7.	Le projet RoboTeach	85
3.7.1.	Caractéristiques générales	85
3.7.2.	Caractéristiques du laboratoire distant	85
3.7.3.	Caractéristiques de l'EIAH	86
3.7.4.	Support à la conception pédagogique.....	86
3.7.5.	Support à l'apprentissage collaboratif en ligne	87
3.7.6.	Support au tutorat en ligne	87
3.7.7.	Synthèse	88
3.8.	Le projet VITELS	89
3.8.1.	Caractéristiques générales	89
3.8.2.	Caractéristiques du laboratoire distant	90
3.8.3.	Caractéristiques de l'EIAH	91

3.8.4.	Support à la conception pédagogique.....	91
3.8.5.	Support de l'apprentissage collaboratif et du tutorat en ligne.....	91
3.8.6.	Synthèse	92
3.9.	Le projet Tele-Lab IT-Security	92
3.9.1.	Caractéristiques générales	92
3.9.2.	Caractéristiques du laboratoire distant	93
3.9.3.	Caractéristiques de l'EIAH	94
3.9.4.	Support à la conception pédagogique.....	94
3.9.5.	Support à l'apprentissage collaboratif en ligne et au tutorat en ligne	95
3.9.6.	Synthèse	95
3.10.	Synthèse générale	95
Chapitre 4.	Les verrous à lever	97
4.1.	L'hétérogénéité des modèles de laboratoires	98
4.2.	L'hétérogénéité des moyens de télé-instrumentation	98
4.3.	La rigidité des modèles d'expériences	98
4.4.	La fédération et la mise en œuvre d'expériences distribuées	99
4.5.	L'hétérogénéité et l'inaccessibilité des traces d'activités	100
4.6.	L'intégration avec les normes et standards pédagogiques existants	101
4.7.	Support à l'expérimentation collaborative et au tutorat en ligne	101
Partie II :	Une architecture et des modèles génériques pour l'intégration des télé-TPs en EIAHs.....	104
Chapitre 5.	Une architecture pour l'intégration des EIAHs et des environnements de télé-TPs	105
5.1.	Une architecture structurée en couches	105
5.2.	La couche d'Apprentissage	108
5.2.1.	L'objet pédagogique « Expérience ».....	110
5.2.2.	Le moteur d'exécution des objets « Expérience ».....	115
5.2.3.	L'IHM de télé-opération	117
5.3.	La couche d'Expérimentation	125
5.3.1.	Architecture des Pilotes de dispositifs	128
5.3.2.	Le protocole de communication avec la couche supérieure.....	129
5.3.3.	Une interface standardisée et unificatrice	129

5.3.4.	La problématique du déploiement automatique des expériences.....	130
5.4.	Synthèse	132
Chapitre 6.	La couche d'Intégration	133
6.1.	Les services offerts à la couche d'Apprentissage	136
6.1.1.	Le service de gestion des laboratoires distants.....	136
6.1.2.	Les services de gestion des expériences en ligne.....	137
6.1.3.	Le service d'authentification et d'autorisation.....	141
6.1.4.	Le service de gestion des traces d'activités.....	142
6.2.	Les composants d'intégration des laboratoires en ligne	143
6.2.1.	Les fondements de notre approche.....	144
6.2.2.	Le serveur WBEM primaire.....	148
6.2.3.	Les serveurs WBEM secondaires.....	152
6.2.4.	Les Pilotes des dispositifs de laboratoires.....	153
6.2.5.	Le protocole d'échange d'informations	153
6.3.	Synthèse	154
Chapitre 7.	Des modèles génériques pour les télé-TPs.....	156
7.1.	La modélisation CIM	157
7.1.1.	Le méta-modèle CIM	157
7.1.2.	Le modèle d'informations CIM.....	160
7.1.3.	CIM comme méta-modèle pour les OPI « Expérience »	162
7.2.	Modélisation du domaine des télé-TPs	162
7.2.1.	Sous-modèle de gestion des laboratoires	163
7.2.2.	Sous-modèle de gestion des expériences	167
7.2.3.	Exemple de modélisation d'une expérience.....	170
7.3.	Modélisation du domaine des EIAHs	171
7.4.	Modélisation des utilisateurs et de leurs activités	175
7.4.1.	Le profil de l'apprenant.....	175
7.4.2.	La représentation des activités	177
7.5.	Synthèse	179
Chapitre 8.	Synthèse des contributions	180
8.1.	Du point de vue technique et technologique	180
8.1.1.	Une modélisation standardisée des télé-TPs	181

8.1.2.	Des interfaces de contrôle et de supervision uniformes	183
8.2.	Du point de vue pédagogique	184
8.2.1.	Intégration et interaction de l'objet « Expérience » avec les EIAHs	184
8.2.2.	Support à la conception d'expériences	185
8.2.3.	Support aux interactions avec les expériences et au tutorat synchrone.....	185
8.2.4.	Des IHMs adaptées.....	185
8.2.5.	Support de l'apprentissage collaboratif et du tutorat asynchrone	186
Partie III : Cas d'étude : l'enseignement des systèmes et réseaux informatiques.....		188
Chapitre 9.	Un EIAH dédié aux travaux pratiques en ligne en informatique	189
9.1.	Introduction au domaine d'enseignement visé	190
9.1.1.	Objectifs pédagogiques globaux d'un cursus informatique	191
9.1.2.	Les travaux pratiques en informatique	192
9.1.3.	Caractéristiques des laboratoires en ligne	192
9.1.4.	Un scénario type de télé-TP	194
9.2.	Méthodologie d'implémentation	197
9.3.	Les modèles CIM spécifiques aux expériences en informatique	198
9.3.1.	Modèle de gestion des ressources de laboratoires.....	199
9.3.2.	Modèle de gestion des expériences	203
9.3.3.	Modèle de l'OPI « Expérience »	208
9.3.4.	Modèle des traces d'activités	209
9.3.5.	Bilan	212
9.4.	Implémentation de la couche d'Apprentissage	213
9.4.1.	La plateforme d'apprentissage	215
9.4.2.	Le moteur d'exécution des expériences	215
9.5.	Implémentation de la couche d'Intégration	218
9.5.1.	Le serveur WBEM OpenPegasus	220
9.5.2.	Les services rendus à la couche d'Apprentissage	220
9.6.	Implémentation de la couche d'Expérimentation	222
9.6.1.	Le niveau inférieur	223
9.6.2.	Le niveau intermédiaire.....	226
9.6.3.	Le niveau supérieur : les Pilotes WBEM	231
9.7.	Scénario de coopération des différentes couches	235

9.8. Synthèse	236
Chapitre 10. Outils et IHMs supports aux activités pédagogiques	238
10.1. Introduction	238
10.2. Outils et IHMs supports aux activités de l'enseignant concepteur	239
10.2.1. Conception des expériences avec un éditeur graphique	240
10.2.2. Traduction vers les formats MOF et MLN.....	243
10.2.3. Gestion du cycle de vie des expériences	245
10.2.4. Stockage et partage des objets pédagogiques « Expérience ».....	249
10.3. Outils et IHMs supports aux activités d'apprentissage individuel	249
10.3.1. La barre de menus	251
10.3.2. L'explorateur des expériences.....	252
10.3.3. Exploitation de l'expérience.....	256
10.3.4. L'explorateur des utilisateurs	259
10.4. Description de l'espace d'apprentissage collaboratif	261
10.4.1. Le sentiment de présence	261
10.4.2. Exploitation de sessions ouvertes par d'autres acteurs	261
10.4.3. Création et exploitation de sessions partagées	262
10.4.4. L'outil de messagerie instantanée	263
10.5. Description de l'espace de travail du tuteur	265
10.6. Synthèse	268
Chapitre 11. Expérimentation	270
11.1. Introduction	270
11.2. Démarche	272
11.3. Test d'utilisabilité	273
11.4. Résultats et interprétations	276
11.4.1. Observation du comportement des utilisateurs	277
11.4.2. Enquête de satisfaction.....	278
11.5. Synthèse	279
Conclusion.....	283
Chapitre 12. Conclusions et perspectives	284
12.1. Bilan général	284
12.1.1. Introduction et normalisation de l'objet pédagogique « Expérience ».....	285

12.1.2.	Originalités et avantages des expériences modélisées en CIM	287
12.1.3.	La couche d'Intégration : un moteur générique pour l'opérationnalisation des expériences	288
12.1.4.	Des modèles génériques pour la gestion des expériences et des traces.	289
12.1.5.	Outils et IHMs supports à l'apprentissage et au tutorat	290
12.2.	Perspectives	291
12.2.1.	Perspectives générales.....	291
12.2.2.	Perspectives spécifiques au domaine informatique.....	295
Chapitre 13.	Bibliographie	299
Annexes	325
Annexe A :	Synthèse de l'état de l'art des projets de télé-TPs existants	326
Annexe B :	Services WEB dédiés à la gestion de laboratoires en ligne	330
Annexe C :	Services WEB dédiés à la gestion des expériences en ligne.....	332
Annexe D :	Conception, construction et déploiement de machines virtuelles.....	335
D.1	Phase de conception	335
D.2	Phase de construction	336
D.3	Phase de déploiement	338
D.4	Phase d'exploitation	338
D.5	Phases d'arrêt et de suspension	338
D.6	Phase de sauvegarde et d'archivage	338
D.7	Phase de suppression	339
Annexe E :	Diagramme de l'expérience type en informatique selon quatre formats	340
Annexe F :	Mode d'emploi de l'interface de Télé-expérience.....	348
Introduction et mise en garde	348
1.	Premier accès et vue d'ensemble	348
2.	La messagerie instantanée	351
a.	Le canal de l'équipe	352
b.	Le canal du tuteur	352
c.	Emoticons et actes de langages	352
3.	La fenêtre d'expérimentation	352
a.	Explorateur de l'expérience	353
b.	Explorateur des utilisateurs	355

4. Une Activité type de Travaux Pratiques en Systèmes et Réseaux : Adressage et Configuration IP de base sous Linux	358
1. Introduction	358
Consignes de travail	358
2. Question 1	359
3. Question 2	359
4. Question 3	360
5. Question 4	360
6. Question 5	361
6. Tests	362
7. Synthèse	362
Annexe G : Résultats résumés de l'Enquête sur l'Utilisabilité de l'Environnement de Télé-TP	363

Liste des tableaux

Tableau 2-1. Objectifs pédagogiques des travaux pratiques.....	42
Tableau 3-1. Catégories des critères d'analyse des projets de télé-TPs.....	66
Tableau 6-1. Spécification des méthodes du service de gestion des laboratoires.....	136
Tableau 6-2. Spécification des méthodes du service de gestion du cycle de vie des expériences.....	138
Tableau 6-3. Spécification des méthodes du service de télé-opération des expériences.....	140
Tableau 6-4. Correspondance entre fonctions d'un OPI « Expérience » et les méthodes spécifiées par les services de gestion des expériences.....	140
Tableau 6-5. Spécification des méthodes du service de gestion des traces d'activités.....	142
Tableau 7-1. Attributs de la classe <i>LAB_Domain</i>	164
Tableau 7-2. Attributs additionnels de la classe <i>LAB_Resource</i>	165
Tableau 7-3. Attributs de la classe <i>LAB_ResourcePool</i>	166
Tableau 7-4. Méthodes de la classe <i>LAB_Experiment</i>	167
Tableau 9-1. Combinaison des valeurs des attributs dédiés à la gestion d'expériences locales ou distribuées.....	204
Tableau 9-2. Comparaison des différentes technologies de virtualisation.....	224
Tableau 9-3. Phases du cycle de vie d'une machine virtuelle.....	225
Tableau 9-4. Projection des critères de sélection sur des outils de gestion d'expériences réseau virtuelles.....	228
Tableau 9-5. Correspondance entre les attributs/méthodes d'un objet <i>LAB_ComputerExperiment</i> et le projet MLN associé.....	232
Tableau 10-1. Exemples d'options de configuration ajoutées à WeatherMap.....	242
Tableau 10-2. Méthodes implémentées pour la gestion du cycle de vie d'une expérience réseau.....	248
Tableau 11-1. Résumé du questionnaire d'enquête sur l'utilisabilité.....	274
Tableau E-1: Quelques instances CIM en format MOF de l'expérience.....	341
Tableau E-2 : Modèle de l'expérience en format MLN.....	343
Tableau E-3 : Modèle de l'expérience format Network WeatherMap.....	345

Liste des figures

Figure 2-1. Interrelation entre expériences, travaux en laboratoires et travaux pratiques (Hodson 1988 ; cité dans Marthie, 1995).....	34
Figure 2-2. Typologie des laboratoires, adaptée de (Benmohamed, 2006).....	36
Figure 2-3. Processus de développement et d'implémentation d'une tâche pratique (Millar et al., 2002 ; cité dans Millar, 2004).	39
Figure 2-4. Le cycle du modèle d'apprentissage par investigation.....	46
Figure 2-5. Architecture générale d'une plateforme de télé-TP (Gustavo et al., 2007).....	60
Figure 3-1. Architecture générale du système PEARL (Pearl, 2000D4.1).	68
Figure 3-2. IHM d'expérimentation sur un système électronique (Ferreira et al., 2002).	70
Figure 3-3. Architecture générale du projet LAB@FUTURE.	73
Figure 3-4. A gauche la structure de l'interface de suivi, à droite l'état global d'avancement des apprenants pour une activité donnée (Guéraud, 2004).	77
Figure 3-5. Architecture générale du projet ICTT@LAB (BenMohammed, 2006).	79
Figure 3-6. Architecture générale du projet eINST.....	82
Figure 3-7. Deux IHMs de pilotage de RoboTeach (Leroux, 1997).	86
Figure 3-8. Le modèle de Suivi d'Activités d'Apprentissage à Distance (SAAD - Desprès, 2001).....	88
Figure 3-9. A gauche l'architecture générale du projet VITELS, à droite un exemple de laboratoire mis en ligne.	90
Figure 3-10. Architecture générale du projet Tele-Lab IT-Security.	93
Figure 5-1. Une architecture pour l'intégration des EIAHs et des télé-TPs.	107
Figure 5-2. Détails de la couche d'Apprentissage.....	109
Figure 5-3. Contenu de l'objet pédagogique « Expérience ».	114
Figure 5-4. Prototype d'IHM de conception d'une expérience en génie électrique.	118
Figure 5-5. Génération du modèle d'une expérience en électronique.....	118
Figure 5-6. Structure générale d'une interface « Cockpit », et son application au domaine de la mécatronique (Gillet et al., 2003).....	120
Figure 5-7. Architecture d'IHM pour l'exploitation des télé-TPs.	123

Figure 5-8. Prototype d'IHM pour une expérience en génie électrique.....	125
Figure 5-9. Coopération entre l'interface utilisateur et les dispositifs de la couche d'Expérimentation dans le domaine du génie électrique.	127
Figure 5-10. Architecture des Pilotes de dispositifs des laboratoires.	129
Figure 5-11. Déploiement d'une expérience dans un laboratoire.	131
Figure 6-1. Détails de la couche d'Intégration.....	135
Figure 6-2. Composants clés de l'architecture WBEM.....	145
Figure 6-3. Expérience centralisée versus expérience distribuée.....	147
Figure 6-4. Exploitation des modèles d'expérience dans les différentes couches.	151
Figure 7-1. Méta-méta-modèle CIM illustré par un diagramme UML.	158
Figure 7-2. Exemple de classe CIM représentée aux formats MOF, UML et XML.	160
Figure 7-3. Le schéma CIM de base (DMTF, 2003).....	161
Figure 7-4. Les modèles communs de CIM (DMTF, 2003).	161
Figure 7-5. Modèle d'allocation des ressources de laboratoires pour les expériences en ligne.	163
Figure 7-6. Diagramme d'objets illustrant l'exemple de l'expérience en génie électrique. ..	170
Figure 7-7. Modélisation du domaine des EIAHs.....	172
Figure 7-8. Modèle UML de la classe <i>TEL_Experiment</i>	174
Figure 7-9. Association entre les classes <i>TEL_Experiment</i> et <i>LAB_Experiment</i>	174
Figure 7-10. Le profil de l'apprenant.	176
Figure 7-11. Modèle de suivi des activités sur les expériences en ligne.....	178
Figure 9-1. Architecture d'un laboratoire en informatique en ligne.	193
Figure 9-2. Scénario d'un télé-TP traitant des réseaux informatiques.	195
Figure 9-3. Description de l'expérience dans le cadre de notre étude.	197
Figure 9-4. Correspondance entre objets pédagogiques, objets CIM et dispositifs de laboratoire.....	199
Figure 9-5. Modèle de gestion des ressources d'un laboratoire informatique.	201
Figure 9-6. Diagramme d'instances CIM de ressources de laboratoire.	202
Figure 9-7. Modèle général d'une expérience réseau.	203
Figure 9-8. Modèle détaillé d'une expérience réseau.	205
Figure 9-9: Diagramme d'instances CIM-UML de notre expérience type.	207

Figure 9-10 : Modèle de l'OPI « Expérience » en informatique, et sa relation avec le modèle de gestion des expériences.	208
Figure 9-11 : Diagramme d'instances de l'OPI « Expérience » correspondant au scénario type.	209
Figure 9-11. Modèle spécifique pour le suivi des activités de type ligne de commande.....	211
Figure 9-12. Diagramme d'instances d'une session sur un nœud de l'expérience.	212
Figure 9-13. Composants de la couche d'Intégration.	219
Figure 9-14. Niveaux et composants de la couche d'Expérimentation.....	223
Figure 9-15. Coopération des différents outils pour contrôler une machine de l'expérience.	236
Figure 10-1. Processus de prototypage d'une expérience.	240
Figure 10-2. Conception de notre expérience réseau avec Network WeatherMap.....	241
Figure 10-3. Boîte de dialogue des propriétés d'un nœud.	242
Figure 10-4. Machines virtuelles nécessaires à notre expérience.	247
Figure 10-5. Mise en place de la topologie de notre expérience.....	247
Figure 10-6. Une IHM pour le support des activités d'apprentissage.....	251
Figure 10-7. Détails de la barre de menus de l'IHM d'exploitation d'une expérience réseau.	251
Figure 10-8. L'explorateur de l'expérience.....	253
Figure 10-9. Menu contextuel associé à un nœud de l'expérience.	255
Figure 10-10. Menu contextuel associé à une session ouverte sur un nœud de l'expérience.	256
Figure 10-11. Navigation entre différentes sessions Terminal.	257
Figure 10-12. Recherche d'éventuelles sessions existantes et restauration d'une session retrouvée.....	258
Figure 10-13. Structure de l'explorateur des utilisateurs et correspondance des éléments affichés avec les instances du modèle d'activités.	259
Figure 10-14. Exploration de ses propres sessions à partir de l'explorateur des utilisateurs.	260
Figure 10-15. Accès partagé à une session ouverte en mode lecture et écriture.....	263
Figure 10-16. Détails de la fenêtre de messagerie instantanée.	264

Introduction

Chapitre 1. Introduction et problématiques abordées

<u>1.1.</u>	<u>Cadre scientifique</u>	21
<u>1.2.</u>	<u>Contexte</u>	22
<u>1.3.</u>	<u>Problématiques</u>	24
<u>1.3.1.</u>	<u>Efficacité pédagogique</u>	25
<u>1.3.2.</u>	<u>Intégration dans les EIAHs</u>	26
<u>1.3.3.</u>	<u>Support des fonctions pédagogiques</u>	26
<u>1.3.4.</u>	<u>L'IHM et la télé-opération</u>	27
<u>1.3.5.</u>	<u>Mutualisation des ressources des laboratoires</u>	27
<u>1.3.6.</u>	<u>Standardisation des expériences</u>	28
<u>1.4.</u>	<u>Organisation de ce manuscrit</u>	28

1.1. Cadre scientifique

Les travaux de recherche présentés dans cette thèse ont été réalisés au sein de l'Institut de Recherche en Informatique de Toulouse (IRIT)¹, et plus précisément au sein de l'équipe SIERA² (*Service Integration and Network Administration*) qui couvre le thème de recherche « Architectures, Systèmes et Réseaux »³.

Les travaux de notre équipe visent « le contrôle et la maîtrise des infrastructures et services de communication de dernières générations, mais aussi des systèmes et applications complexes dynamiquement agrégés, distribués et trans-organisationnels ». Les domaines d'application visés concernent en particulier les réseaux, les services et la e-Formation (ou e-Learning). Les travaux présentés dans ce document s'intègrent dans ce dernier domaine d'application, et

¹ <http://www.irit.fr>

² <http://www.irit.fr/-Equipe-SIERA->

³ <http://www.irit.fr/-Theme-6-Architecture-systemes-et->

s'intéressent en particulier à la conception, la réalisation et les usages des Environnements Informatiques pour l'Apprentissage Humain (EIAH). Dans le but d'aboutir à une distribution à grande échelle d'objets pédagogiques en s'affranchissant de l'hétérogénéité des systèmes, notre équipe s'attache à trouver des solutions visant l'interopérabilité entre entités de stockage des éléments pédagogiques (appelées LOR : *Learning Objects Repository*) et interfaces utilisateurs (appelées LMS : *Learning Management Systems*) au travers d'une architecture fédératrice s'appuyant sur des standards et assurant la gestion du cycle de vie de ces entités.

Afin de faciliter le processus de personnalisation et de réingénierie aussi bien des objets pédagogiques que des interfaces utilisateurs, notre équipe s'intéresse à l'observation, l'instrumentation et l'évaluation de ces systèmes éducatifs dans l'objectif de fournir des enseignements à distance de qualité. Les travaux menés sur cette problématique ont abouti d'un côté à l'élaboration d'un modèle générique qui structure et apporte une sémantique claire aux traces d'activités laissées par les utilisateurs, et d'un autre côté à une architecture dédiée à la collecte et au partage de ces traces qui en facilite l'exploitation et l'échange par les différents acteurs de la formation (Broisin et Vidal, 2007).

1.2. Contexte

Les avancées de la recherche et développement dans le domaine du e-Learning tant aux niveaux théorique ou pédagogique que technique ont contribué au renforcement de sa position en complément de l'enseignement traditionnel. Les avancées technologiques et la généralisation de l'accès à Internet, mais aussi la demande de plus en plus forte pour une formation tout au long de la vie, accentuent l'usage des Technologies de l'Information et de la Communication pour l'Enseignement (TICE) au sein des institutions d'enseignement.

On a dû attendre que les plateformes d'enseignement (en ligne) puissent atteindre un minimum de maturité pour envisager la mise en ligne des travaux pratiques (ou télé-TPs), qui sont caractérisés par la complexité de mise en œuvre des expériences fondées sur l'exploitation de laboratoires. Un engouement particulier a alors émergé ces dix dernières années quant à ce type d'activité, notamment en raison des enjeux pédagogiques et économiques qu'elle engendre (Gravier, 2008). De plus, les travaux pratiques sont des enseignements indispensables aux disciplines scientifiques et technologiques du fait que certains objectifs d'apprentissage, notamment ceux liés à l'acquisition de compétences

professionnelles et scientifiques, de travail de groupe et de communication, ne peuvent être atteints qu'à travers ce genre d'activités. Cet intérêt est justifié par :

- Le besoin de mettre en ligne des cursus de formation dans les domaines de la science, des mathématiques, de l'ingénierie et de la technologie (Lelevé et al., 2002).
- Les évolutions de l'organisation des entreprises dans le domaine industriel (dont les stratégies d'expansion géographique correspondent à une délocalisation des unités de production vers les zones ou pays offrant un marché attractif à travers une main d'œuvre spécialisée et qualifiée à bas coût) ont fait émerger le besoin de télé-instrumenter des dispositifs industriels de production à distance en considérant toutes les implications sociales et humaines de cette médiatisation (Gravier, 2007).

Les recherches en télé-TPs, appliquées initialement aux domaines d'enseignement dont la nature est favorable à la mise en ligne de dispositifs de laboratoire (notamment l'informatique, l'électronique, l'automatique, la robotique, etc.), ainsi que les simulations informatiques, ont permis de mettre en évidence les problèmes pédagogiques et techniques qui entravent le développement et l'acceptation des télé-TPs par les acteurs de la formation : d'un point de vue pédagogique, il est difficile de s'assurer que les télé-TPs garantissent un enseignement efficace à cause du facteur de la distance et des problèmes classiques d'isolement de l'apprenant. D'un point de vue technique, et en dehors des problèmes de gestion des dispositifs physiques à distance, de leur partage et de leur sécurité entre autres, les solutions proposées peinent à s'intégrer efficacement avec les autres enseignements en ligne pour former des cursus complets et homogènes liant efficacement la théorie à la pratique.

Notre contribution enrichit ces travaux par l'intégration d'objets pédagogiques interactifs particuliers que sont les expériences (en laboratoires) en ligne, qui s'insèrent dans des activités d'apprentissage particulières : les travaux pratiques en ligne. Nous nous attachons au cours de ce travail à reprendre et étendre les résultats de recherche de notre équipe afin d'offrir les mêmes services de fédération et de virtualisation des expériences (Broisin, 2006 ; Catteau, 2008), ainsi que l'observation des activités sur ces ressources tout en proposant des extensions spécifiques aux travaux pratiques en ligne. Les travaux menés dans le cadre de cette thèse tentent de trouver un compromis entre les développements récents, tant théoriques que techniques, sur les environnements informatiques pour l'apprentissage humain, les

fondements théoriques des travaux pratiques traditionnels ainsi que ceux des travaux pratiques en ligne, et les avancées technologiques en terme de mise en ligne de dispositifs de laboratoires.

Bien que notre contribution vise à proposer des solutions génériques à tous les domaines d'enseignement concernés par les travaux pratiques, notre expérimentation se focalise sur notre domaine d'enseignement qu'est l'informatique. Ce domaine est celui qui bénéficie le plus des avancées technologiques relatives à la mise en ligne d'expériences, grâce notamment aux technologies de virtualisation⁴ et infrastructures de nuages informatiques (appelées *Cloud Computing*) qui favorisent la mise en place de laboratoires informatiques proches du réel à très faible coût.

Etant donnée la diversité des termes et concepts utilisés dans la littérature, nous ferons tout au long de ce manuscrit la nette distinction entre :

- Le télé-TP qui représente une activité (humaine) d'apprentissage.
- Le laboratoire qui est un espace technique spatial et temporel d'hébergement de dispositifs nécessaires à la conduite des expériences.
- L'expérience qui définit les conditions dans lesquelles se déroule une activité d'expérimentation donnée, conçue et amorcée par un enseignant concepteur, accompagnée par un tuteur, et conduite par les apprenants prenant part au télé-TP.

Les termes « laboratoire virtuel », « laboratoire en ligne » et « laboratoire distant » feront référence aux laboratoires accessibles via un EIAH quelconque ; ils incluent donc les simulations logicielles et les laboratoires physiques accessibles à distance.

1.3. Problématiques

Les avancées technologiques relatives aux dispositifs de laboratoires télé-opérables via un réseau ont accru les pressions sur les universités pour qu'elles assurent la transition des laboratoires classiques vers les laboratoires virtuels (Ma et Nickerson, 2006) ; il s'agit essentiellement de pressions économiques liées aux coûts des espaces alloués aux laboratoires, aux équipements et à leur maintenance. Mais cette transition ne peut réussir sans

⁴ Nous entendons par technologies de virtualisation, les technologies qui permettent d'exécuter plusieurs machines virtuelles au sein d'une même machine physique (<http://fr.wikipedia.org/wiki/Virtualisation>).

la prise en charge de certains aspects pédagogiques qui constituent les éléments de notre problématique : assurer l'efficacité pédagogique des télé-TPs en considérant les théories d'apprentissage, intégrer les laboratoires distants au sein des EIAHs disponibles aujourd'hui, supporter les fonctions pédagogiques de conception, de tutorat et bien entendu d'apprentissage, proposer une Interface Homme Machine (IHM) de télé-opération simple qui offre les fonctions pédagogiques décrites ci-dessus, mutualiser les ressources physiques et logicielles afin de permettre aux apprenants d'accéder à d'autres équipements (et donc à d'autres connaissances), et enfin normaliser les ressources pédagogiques associées aux télé-TPs afin de permettre leur partage et réutilisation à grande échelle.

1.3.1. Efficacité pédagogique

L'accélération de la mise en ligne des travaux pratiques via les laboratoires virtuels a divisé la communauté des enseignants et pédagogues sur l'efficacité pédagogique de ce mode d'enseignement (Ma et Nickerson, 2006 ; Corter et al., 2007 ; Nickerson et al., 2007). Alors que les défenseurs des travaux pratiques en ligne mettent en avant la levée des contraintes liées aux laboratoires traditionnels tout en offrant de nouvelles possibilités techniques et pédagogiques (Lelevé, 2002), les détracteurs avancent que les objectifs d'un cursus dans le domaine de la science, de l'ingénierie ou de la technologie ne peuvent être atteints sans que l'étudiant ne soit confronté directement à des situations d'expérimentations réelles pendant sa formation. La réussite des télé-TPs est donc conditionnée par leur efficacité pédagogique, c'est-à-dire l'atteinte des résultats obtenus et engendrés par les travaux pratiques traditionnels (Ma et Nickerson, 2006 ; Nickerson et al., 2007).

Les recherches comparatives entre les deux types de travaux pratiques insistent sur la nécessité d'explorer l'impact de la médiatisation des expériences sur l'apprentissage afin d'identifier les bonnes pratiques d'enseignement. Pour atteindre l'efficacité pédagogique souhaitée, nous pensons que la problématique réside dans l'application des éléments des différentes théories d'apprentissage, en particulier le socioconstructivisme et le cognitivisme.

Par ailleurs, et dans la perspective de formations complètement à distance, les activités de télé-TPs doivent couvrir toutes les situations d'expérimentation possibles sans être restreintes à la télé-opération de dispositifs distants sur laquelle se focalise la plupart des travaux existants. En effet, le rapport de (Singer et al., 2005) recommande d'autres types

d'expériences telles que l'émission de questions et d'hypothèses de recherche, la conception d'expériences de recueil de données de mesure, l'exploitation et l'analyse de données d'expérimentations recueillies par des tiers et stockées dans des bases de données, etc. Enfin, les résultats des projets fondés sur les simulations qui ont atteint un certain degré de maturité devront être pris en considération (Pernin, 1996 ; Buitrago, 1998 ; Guerraud, 2005).

1.3.2. Intégration dans les EIAHs

L'introduction des télé-TPs dans la formation à distance ne peut se faire qu'à travers leur intégration transparente au sein des EIAHs existants. Ce manque entraîne en effet un rejet des télé-TPs de la part des acteurs de la e-Formation qui sont souvent familiarisés à un EIAH existant où ils ont assimilé certaines habitudes d'apprentissage et d'enseignement. De plus, les activités de télé-TPs ne peuvent être déconnectées du contexte général d'un cursus de formation, car justement un de leurs objectifs est de lier la théorie à la pratique, de produire de la documentation, d'apprendre en équipe, etc.

Comme nous allons le démontrer plus loin, peu de travaux se sont intéressés à l'intégration des expériences en laboratoires distants dans les EIAHs existants, à l'exception de ceux fondés sur les simulations. Les projets traitant cette problématique sont peu avancés (Benmohamed, 2006 ; Gravier, 2007), puisque la majorité des travaux propose des EIAHs spécifiques constitués de simples IHMs de télé-opération qui sont complètement déconnectées de l'EIAH utilisé pour les autres types d'activité d'apprentissage.

1.3.3. Support des fonctions pédagogiques

En dehors des travaux qui utilisent les simulations, nous avons trouvé peu de travaux supportant l'apprentissage collaboratif et le tutorat synchrone, ce qui constitue pour une formation partiellement ou complètement à distance des écueils dont des solutions adaptées doivent être apportées.

La collaboration a été identifiée comme un élément essentiel pour vaincre l'isolement de l'apprenant qui conduit généralement à l'abandon dans le cadre d'une formation ouverte et à distance (Faerber 2002, 2004). Aussi, dans le cadre des travaux pratiques, la collaboration n'est pas seulement un moyen pédagogique, mais constitue un objectif pédagogique visé (Singer et al., 2005 ; ABET, 2010) ; il est rare qu'un chercheur ou un ingénieur travaille seul

sur un projet. Les quelques travaux qui tentent de couvrir partiellement cet aspect sont présentés dans le chapitre 3.

1.3.4. L'IHM et la télé-opération

L'aspect sur lequel se focalise la majorité des travaux sur les télé-TPs est sans doute celui de l'IHM de télé-opération du laboratoire en ligne. Cette orientation se comprend parfaitement du fait de la grande diversité des dispositifs technologiques utilisés dans les divers domaines d'enseignement, où différents instruments sont nécessaires en fonction de la situation d'apprentissage. Cependant, il s'agit d'un véritable frein au développement des télé-TPs car il est nécessaire, pour chaque nouvel équipement, de développer une IHM spécifique reproduisant son apparence, et dont la télé-commande est médiatisée par un protocole de communication souvent spécifique aussi. En dehors des travaux sur les simulations, quelques travaux ont essayé d'apporter des solutions pour modéliser les laboratoires distants, par exemple à partir d'ontologies. Cette solution très intéressante n'assure pas, pour le moment, une séparation claire entre le modèle et le composant logiciel qui l'opérationnalise, puisque ce dernier est intégré dans l'IHM de télé-opération. Par ailleurs, ces solutions ne prennent pas en compte quelques aspects importants de l'apprentissage en ligne comme l'expérimentation collaborative synchrone, l'adaptation aux objectifs de l'activité et aux profils des apprenants (débutants, avancés, handicapés, etc.), ou le suivi des activités des apprenants au cours d'un télé-TP.

1.3.5. Mutualisation des ressources des laboratoires

Même si les ressources des laboratoires sont accessibles en ligne, elles répondront difficilement à la forte demande et au nombre élevé d'apprenants. Cela implique la nécessité de partager et de mutualiser ces ressources entre universités et organismes de formation et de recherche (Zimmerli et al., 2003). Or le partage et la mutualisation des ressources supposent que toutes les plateformes parlent le même langage de communication, offrent les mêmes services et échangent les mêmes formats de données avec les EIAHs ; dans le cas des expériences en ligne, les plateformes doivent également implanter le même langage de commande. Une standardisation de la gestion des plateformes distantes de laboratoires et du protocole de communication associé est donc nécessaire.

1.3.6. Standardisation des expériences

Enfin, l'échange et le partage des expériences entre les différents acteurs de la formation (administrateurs, enseignants, tuteurs et apprenants) nécessitent la standardisation des formats de description des expériences et des métadonnées associées. Si nous considérons l'expérience comme un objet pédagogique, il est nécessaire qu'elle respecte un format de description normalisé permettant son exécution dans n'importe quelle plateforme. Dans le contexte de la formation à distance, une description fondée sur les standards de métadonnées tels que le *Learning Object Metadata* (LOM) devrait lui être associée afin de pouvoir l'intégrer dans des dépôts de ressources pédagogiques aux côtés des cours, exercices, questionnaires, etc. Il serait alors possible de facilement retrouver et intégrer une expérience dans un scénario d'apprentissage ou une séquence de formation.

1.4. Organisation de ce manuscrit

Dans ce premier chapitre, nous avons explicité le cadre scientifique et le contexte de ces travaux avant d'énoncer les problématiques générales liées à la mise en ligne de travaux pratiques. Nous présentons dans le chapitre qui suit les activités de travaux pratiques d'une manière générale, et explicitons les caractéristiques cruciales qui déterminent leur efficacité pédagogique. Aussi nous exposons ces activités dans un contexte à distance, où les spécificités des laboratoires en ligne et les critères à mesurer pour atteindre l'efficacité pédagogique souhaitée sont discutés. Leurs positionnement et intégration dans les environnements informatiques pour l'apprentissage humain, en particulier vis-à-vis des normes et standards pédagogiques, sont également étudiés.

Dans le troisième chapitre, nous introduisons notre modèle d'analyse de l'existant (fondé sur les éléments théoriques du deuxième chapitre) afin d'analyser huit EIAHs dédiés aux télé-TPs. Nous avons éliminé de cette étude les projets dont l'objectif principal est de faciliter la télé-opération de laboratoires en ligne, du fait qu'ils se limitent aux aspects techniques et technologiques, et ignorent les aspects pédagogiques. Une synthèse exposant les points forts et faiblesses des travaux existants clôturera ce chapitre. Dans le chapitre 4, nous explicitons en détail les problématiques liées aux télé-TPs que nous essaierons de solutionner dans le cadre de ce doctorat. Notons que toutes les problématiques identifiées tournent autour de l'atteinte

de l'efficacité pédagogique des télé-TPs, que ce soit du point de vue technique et technologique que du point de vue enseignement et apprentissage.

Les chapitres 5, 6, 7 et 8 constituent la deuxième partie de ce manuscrit, et introduisent notre approche pour mettre en ligne de manière efficace des activités de travaux pratiques. Nous présentons dans le chapitre 5 notre architecture globale structurée en trois couches en détaillant les couches supérieures (les EIAHs) et inférieures (les laboratoires) ; le chapitre 6 se focalise sur la couche intermédiaire qui assure la liaison transparente entre EIAHs et laboratoires en ligne en s'appuyant sur un puissant système d'informations lié à un standard de gestion de réseaux et de systèmes. Notre approche étant fondée sur la décomposition du télé-TP en différents composants élémentaires, le chapitre 7 introduit différents méta-modèles génériques qui, d'une part, décrivent de façon formelle les éléments inhérents à l'activité de télé-TP (laboratoires, dispositifs et expériences) ainsi que les activités réalisées par les utilisateurs sur ces entités, et d'autre part dirigent le contrôle et la supervision des expériences et laboratoires. Le chapitre 8 clôture cette partie en exposant une synthèse générale de nos contributions tant des points de vue technique et technologique que pédagogique.

La troisième et dernière partie de ce manuscrit suggère une application de notre solution théorique à notre domaine d'enseignement, celui des systèmes et réseaux informatiques. Dans le chapitre 9, nous présentons notre implémentation de l'architecture globale proposée dans la deuxième partie, alors que le chapitre 10 se concentre sur différents outils et interfaces spécifiques à chaque acteur du télé-TP (enseignant-concepteur, tuteur, apprenant et administrateur) ; ces IHMs sont caractérisées par un équilibre entre d'un côté l'ergonomie, l'accessibilité et la simplicité de l'usage, et de l'autre la complexité des tâches et la richesse des fonctionnalités. Nous présentons dans le chapitre 11 les résultats d'une expérimentation menée avec deux équipes d'apprenants qui reconforte nos différentes propositions.

Enfin, le chapitre 12 achève ce manuscrit en exposant nos conclusions ainsi que les perspectives futures relatives aux télé-TPs d'une manière générale et aux télé-TPs en informatique en particulier.

Partie I : Etat de l'art

Chapitre 2. Les travaux pratiques et les travaux pratiques à distance pour l'apprentissage de la science

<u>2.1.</u>	<u>Introduction</u>	32
<u>2.2.</u>	<u>Caractérisation des travaux pratiques et télé-TPs</u>	33
<u>2.2.1.</u>	<u>Définitions des TP</u>	33
<u>2.2.2.</u>	<u>Définition des laboratoires</u>	35
<u>2.2.3.</u>	<u>Définition des expériences</u>	36
<u>2.3.</u>	<u>Dimensions d'analyse de l'efficacité pédagogique</u>	39
<u>2.3.1.</u>	<u>Les objectifs pédagogiques</u>	40
<u>2.3.2.</u>	<u>La conception pédagogique</u>	43
<u>2.3.3.</u>	<u>Les activités d'apprentissage</u>	47
<u>2.3.4.</u>	<u>La présence</u>	48
<u>2.3.5.</u>	<u>La fidélité de la télé-instrumentation</u>	49
<u>2.3.6.</u>	<u>L'apprentissage collaboratif</u>	50
<u>2.3.7.</u>	<u>Le tutorat</u>	52
<u>2.3.8.</u>	<u>L'Awareness</u>	53
<u>2.4.</u>	<u>Intégration des télé-TPs en EIAH</u>	56
<u>2.4.1.</u>	<u>Introduction aux EIAH</u>	56
<u>2.4.2.</u>	<u>Les standards pédagogiques</u>	58
<u>2.4.3.</u>	<u>Positionnement des télé-TPs</u>	58
<u>2.5.</u>	<u>Synthèse</u>	62

2.1. Introduction

La communauté des chercheurs et éducateurs admet le rôle central et distinctif que jouent les travaux pratiques dans l'enseignement des sciences (Hofstein et Lunetta, 2003). Ils favorisent l'acquisition de connaissances scientifiques, et encouragent le développement de diverses compétences et habilités scientifiques, techniques et sociales, ainsi que des compréhensions conceptuelles (SCORE, 2009). Les travaux pratiques constituent une composante essentielle

dans l'enseignement de la science car ils contribuent à en atteindre des buts et compétences précis, en complément des autres types d'enseignement (Millar, 2004). Pour ce dernier auteur, l'intérêt majeur des TP est de permettre aux apprenants de faire le lien entre deux domaines : le domaine des objets et faits concrets, et le domaine des idées. Ils visent également à développer et ancrer le raisonnement scientifique chez les apprenants.

L'objectif de ce chapitre est de caractériser les activités d'apprentissage de type travaux pratiques, à la fois dans le cadre d'une formation traditionnelle et dans celui d'une formation ouverte et à distance. Nous nous focaliserons en particulier sur les objectifs pédagogiques visés par les TP qui vont guider leur conception, mise en œuvre et évaluation. Nous discuterons également des catégories de TP et des expériences qui y sont associées. Enfin, nous exposerons les éléments qui permettent d'atteindre l'efficacité pédagogique de ces activités. Nous nous intéresserons de près à leur intégration dans une démarche de conception pédagogique qui repose sur les théories et modèles d'apprentissage modernes, afin de mettre en exergue le potentiel de l'apprentissage par investigation, de l'apprentissage collaboratif, du tutorat et de l'évaluation dans ce contexte.

Cette analyse nous permettra dans la suite du manuscrit de préciser le cadre théorique qui fonde l'analyse de l'existant, et de dégager les principaux éléments théoriques, méthodologiques, pédagogiques et techniques qui vont guider les processus de conception et de construction d'un EIAH dédié aux télé-TPs.

2.2. Caractérisation des travaux pratiques et télé-TPs

2.2.1. Définitions des TP

Nous avons trouvé différentes définitions dans la littérature pour les activités de travaux pratiques, et constaté une diversité des termes utilisés pour en faire référence. Tout comme (Millar, 2004) et (Marthie et Kirshcner, 1995), nous retenons tout au long de ce manuscrit le terme « *Travaux Pratiques* » ou son abrégé « *TP* » pour faire référence à ce genre d'activités car les autres dénominations n'en couvrent qu'une facette :

- Le terme « travaux en laboratoire » dénote seulement les activités qui prennent place dans un endroit dédié à des moments et horaires précis, alors qu'un TP est indépendant de ces deux paramètres et peut s'effectuer chez soi ou ailleurs sans contrainte de temps.

- Le terme « Expérimentations en laboratoire » dénote une forme particulière des travaux pratiques qui consiste à tester une (ou des) hypothèse(s) posée(s) via la conduite d'une ou plusieurs expériences, alors qu'il en existe d'autres formes.
- Le terme « Expériences en laboratoire » dénote les conditions sous lesquelles une expérimentation est menée.

Ces trois aspects sont interdépendants selon une relation hiérarchique schématisée par (Hodson, 1988) et illustrée par la Figure 2-1. Nous avons alors trouvé différentes définitions des TP dans la littérature (Millar, 2004 ; Marthie et Kirshcner, 1995 ; Singer et al., 2005 ; Lunetta et al., 2007), celle que nous avons retenue est la suivante (Millar, 2004) :

« Toute activité d'enseignement et d'apprentissage qui engage à un moment donné les apprenants à observer et manipuler des objets réels et matériels »

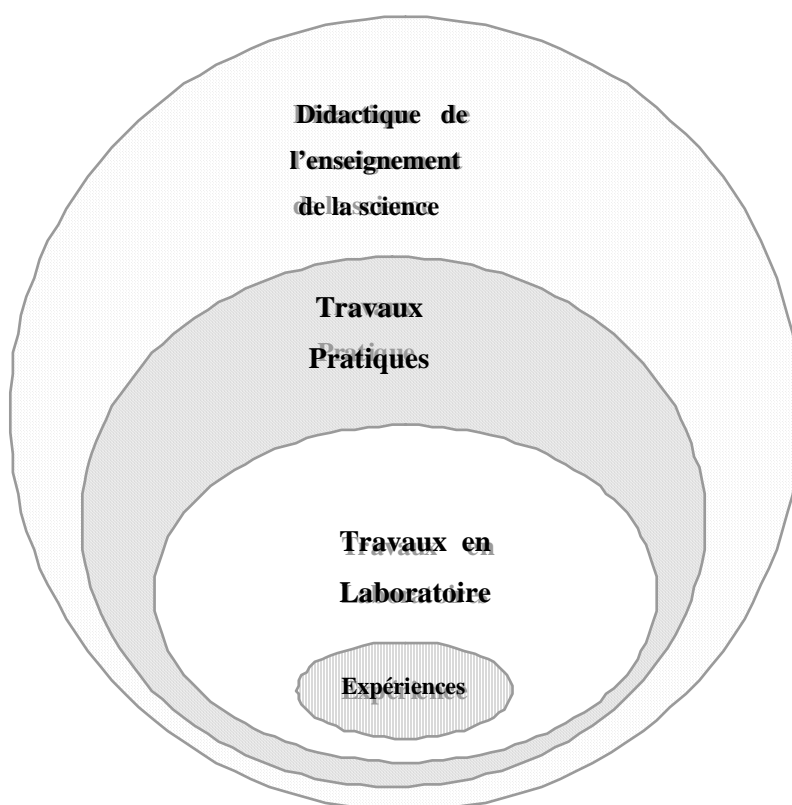


Figure 2-1. Interrelation entre expériences, travaux en laboratoires et travaux pratiques (Hodson 1988 ; cité dans Marthie, 1995).

Nous employons le terme télé-TP par analogie au terme TP, sachant que d'autres termes peuvent être trouvés dans la littérature : TP distant ou TP virtuel. Les télé-TPs sont des activités de travaux pratiques traditionnelles étendues et modifiées pour être accessibles en ligne (Lelevé, 2002 ; Benmohamed, 2006).

2.2.2. Définition des laboratoires

Selon le dictionnaire libre⁵, un laboratoire est défini comme «

- Un local ou un bâtiment équipé pour des expériences scientifiques ou de recherche.
- Une période académique consacrée à travailler ou à étudier dans un tel lieu.
- Un lieu de pratique, d'observation ou d'essai. »

Le terme laboratoire est souvent utilisé dans la littérature pour désigner le(s) dispositif(s) technologique(s) utilisé(s) pour la conduite des expériences ; ceci s'applique aux travaux pratiques à distance puisque l'apprenant distant n'est pas physiquement dans le « laboratoire », mais dispose d'outils informatiques lui permettant de télécommander les dispositifs qui se trouvent dans ce laboratoire. (BenMohammed, 2006) identifie trois grandes catégories de laboratoires schématisées sur la Figure 2-2, les sous-catégories étant projetées sur un continuum à deux extrémités (réel et virtuel) :

- Les laboratoires physiques regroupent tous les dispositifs technologiques physiques.
- Les laboratoires virtuels comprennent tous les dispositifs non physiques.
- Les laboratoires hybrides regroupent tous les dispositifs composés d'une partie physique et d'une partie virtuelle, et ne doivent pas être confondus avec les dispositifs physiques commandés par ordinateurs (Benmohamed, 2006).

Quant au laboratoire en ligne, nous le définissons comme un laboratoire classique étendu et modifié pour être télé-opéré via une interface homme-machine. Les laboratoires sont plus ou moins complexes à mettre en ligne selon leur nature : les plus simples sont les laboratoires virtuels (simulations) et émulés, les plus compliqués étant les laboratoires physiques nécessitant des dispositifs supplémentaires pour leur médiatisation. Entre ces deux extrêmes,

⁵ <http://www.thefreedictionary.com/laboratory>

certains dispositifs sont médiatisés par ordinateur alors que d'autres embarquent un mini système informatique qui les rend accessibles à distance sans intermédiaire.

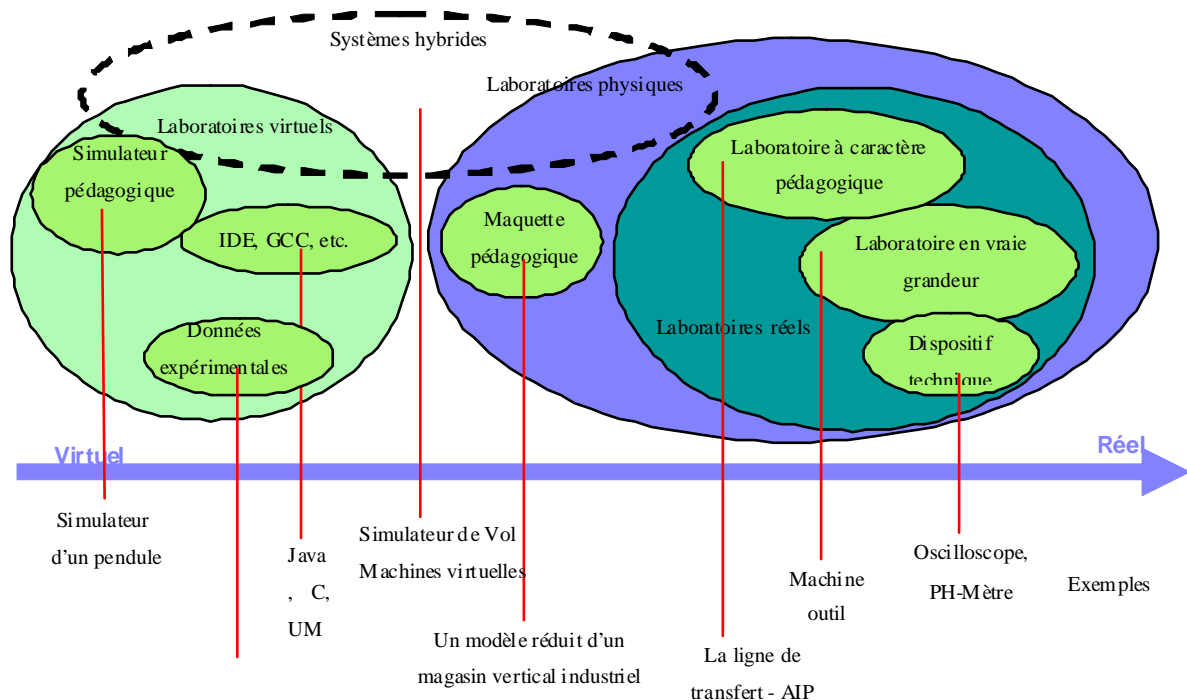


Figure 2-2. Typologie des laboratoires, adaptée de (Benmohamed, 2006).

Aussi, nous classifions les laboratoires en ligne de la même manière que les laboratoires traditionnels, c'est-à-dire dans le continuum Physique-Virtuel-Hybride, puisque la seule différence est le mode d'opération (directe ou à distance).

2.2.3. Définition des expériences

Nous définissons une expérience comme l'ensemble des conditions et paramètres de démarrage d'un laboratoire qui permettent de faire une expérimentation⁶. Ces conditions et paramètres peuvent évoluer librement ou de façon contrôlée, et peuvent être réajustés par l'enseignant et/ou l'apprenant. Enfin, ils peuvent être plus ou moins ouverts selon le degré de liberté donné aux apprenants qui dépend des objectifs du TP, du niveau des apprenants, etc. Cette définition sous-tend les possibilités de conduire des expériences différentes en utilisant le même laboratoire, et d'inclure différentes expériences dans une même activité d'apprentissage exploitant un ou plusieurs laboratoires.

⁶ <http://www.thefreedictionary.com/experiment>

Les caractéristiques techniques des expériences dépendent des propriétés du laboratoire, ce qui nous amène à catégoriser les expériences de manière similaire. Nous distinguons toutefois les expériences simulées et les expériences émulées qui prennent place dans des laboratoires virtuels :

- Les expériences réelles s'effectuent dans des conditions réelles. Les intérêts pédagogiques de ce type d'expériences sont multiples : renforcement de la motivation, acquisition de compétences et habilités techniques et scientifiques, ou conduite d'expériences réalistes. Mais elles souffrent de certains problèmes liés à la sécurité des personnes ou à l'observation de tous les apprenants et de tous les phénomènes visés par l'expérience aux moments voulus.
- Les expériences simulées s'effectuent dans des conditions idéales, contrôlables et répétables, dont l'évolution est prédictible. Les intérêts pédagogiques de telles expériences sont la maîtrise efficace de certaines habilités, la maîtrise du phénomène simulé, l'observation de phénomènes difficiles à prévoir sur des dispositifs physiques, ou le droit à l'erreur. Les inconvénients des simulations se limitent à la perte du contexte et à la difficulté de reconnaître le phénomène « dans la nature ».
- Les expériences émulées prennent place dans des environnements techniques non physiques, mais qui reproduisent le même comportement des systèmes physiques, à la limite entre les simulations et les systèmes réels. Ce genre d'expériences allie le réalisme des systèmes physiques et la plupart des propriétés des simulations, à l'exception de la prédictibilité des événements. Le meilleur exemple est la virtualisation des systèmes informatiques qui permet d'exécuter des machines virtuelles au sein d'un système informatique physique.
- Les expériences hybrides sont réalisées sur une combinaison de systèmes réels, simulés et émulés, permettant de combiner les avantages de chacun selon la situation.

Par analogie au laboratoire en ligne, nous définissons une expérience en ligne comme une expérience traditionnelle étendue et modifiée pour être conduite en ligne. Nous pensons que la notion d'expérience prend tout son sens dans le contexte de la formation à distance pour différentes raisons :

- Pour le même laboratoire en ligne, différentes IHMs (ou différentes configurations d'une même IHM) sont proposées selon le profil de l'utilisateur et l'activité qu'il réalise.

L'expérience peut être modélisée de sorte que certains aspects du laboratoire sont cachés alors que d'autres sont mis en évidence afin de guider l'apprenant.

- L'expérience en ligne est complètement indépendante du laboratoire, elle peut prendre place dans n'importe quel laboratoire distant. La construction d'expériences plus riches et complexes fondées sur plusieurs laboratoires en ligne est même envisageable, ce qui n'est pas possible avec les TPs traditionnels ; de tels dispositifs sont déjà implémentés dans des projets de collaboratoires (Agarwal et Johnston, 1998), de grilles de calcul distribuées (Johnston, 2002 ; Hoffmann, 2005 ; Jones, 2005), ou de bancs d'essais comme NetBed-Emulab (Brian et al., 2002) et PlanetLab (Fiuczynski, 2006).

Cependant, même si les expériences en ligne présentent de nombreux avantages, quelques faiblesses persistent, principalement liées à l'acceptation par les apprenants et enseignants de ce mode d'expérimentation.

2.3. Dimensions d'analyse de l'efficacité pédagogique

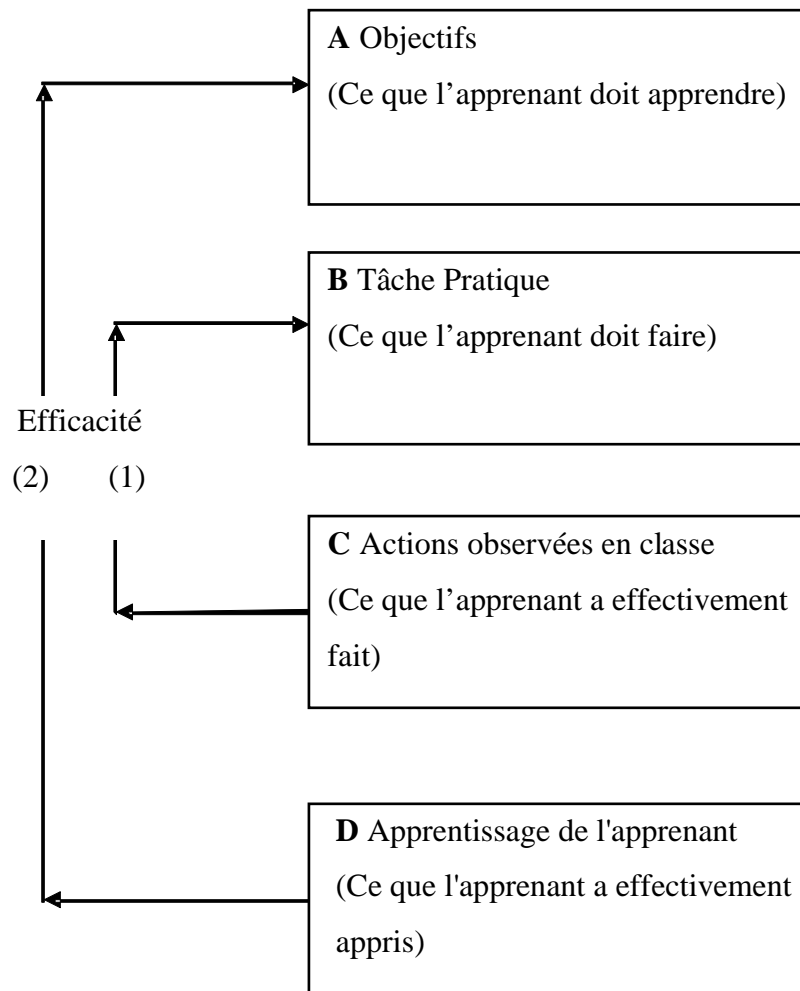


Figure 2-3. Processus de développement et d'implémentation d'une tâche pratique (Millar et al., 2002 ; cité dans Millar, 2004).

L'efficacité pédagogique d'une activité d'apprentissage (ou la réponse à la question « est-ce qu'il y a eu apprentissage ? ») peut être mesurée de deux manières (Ma et Nickerson, 2006) : l'écart entre les résultats obtenus à la fin de l'activité et les objectifs pédagogiques définis au départ, et le calcul du même écart, mais en attribuant cette fois des poids différents à chaque objectif. L'évaluation de cette efficacité est un peu plus compliquée lorsqu'il s'agit d'activité d'expérimentation, car l'apprenant peut faire preuve de maîtrise des tâches demandées sans nécessairement avoir compris le phénomène (Millar, 2004 ; Singer et al., 2005).

(Millar, 2004) définit l'efficacité pédagogique d'une tâche pratique selon deux niveaux : ce que l'apprenant « doit faire », et ce que l'apprenant « doit apprendre ». Il schématise le processus de développement et d'implémentation d'une tâche pratique comme un cycle à nombre fini d'itérations, qui s'arrête lorsque les deux objectifs sont atteints (voir Figure 2-3). Autrement dit, l'auteur considère que l'apprentissage dans le cadre d'un TP est effectif lorsque les apprenants ont compris pourquoi ils ont fait ce qu'ils ont fait, et non pas seulement lorsqu'ils ont réalisé ce qu'il leur était demandé.

Peu de recherches s'intéressent à l'efficacité pédagogique des télé-TPs. Certains travaux (Ma et Nickerson, 2006 ; Nickerson et al., 2007 ; Corter et al., 2007 ; Zacharia et Olympiou, 2010) débattent de l'efficacité de chaque type de laboratoire (d'un point de vue pédagogique, économique, etc.), mais les protagonistes ne proposent pas de critères précis pour une évaluation objective de l'efficacité pédagogique de chaque type de télé-TPs.

Dans les sections ci-après, nous synthétisons un grand nombre de dimensions d'analyse tirées de différents travaux de recherche théoriques et empiriques qui s'attachent à identifier les critères à considérer pour favoriser l'atteinte de l'efficacité pédagogique.

2.3.1. Les objectifs pédagogiques

2.3.1.1. Introduction aux objectifs pédagogiques

Le point de départ dans la conception de toute activité d'apprentissage est la définition des objectifs pédagogiques attendus de cette activité (Joanna, 1996). Plus les objectifs sont définis précisément, plus la conception de l'activité est encadrée et la communication avec les autres acteurs facilitée. De plus, il a été démontré que la présentation de ces objectifs au début d'une activité exerce une fonction pédagogique de formation et d'orientation qui augmente la motivation des apprenants, et donc améliore l'efficacité de leur apprentissage (Depover, 2006c).

Un objectif d'apprentissage est défini par ce qu'un « apprenant sera capable de faire » après une session d'apprentissage, et qu'il ne pouvait pas faire avant (Mager, 1962 ; cité dans Joanna, 1996). Il existe deux types d'objectifs : les objectifs généraux et les objectifs spécifiques. Un objectif général est un « énoncé d'intentions pédagogiques » à partir

desquelles des objectifs spécifiques vont découler, exprimés en termes d'actions attendues des apprenants qui sont observables et mesurables.

Les résultats d'apprentissage d'une activité sont alors évalués par la mesure de l'écart entre les objectifs prédéfinis (actions attendues) et les résultats obtenus (actions réalisées). La meilleure façon d'explicitier les objectifs pédagogiques est de se conformer à une taxonomie comme celle de Bloom ou une de ses dérivées. La taxonomie de Bloom (Bloom, 1956) répartit les objectifs pédagogiques en trois domaines divisés en catégories (Clark, 2009) :

- Le domaine cognitif concerne la connaissance et le développement des habiletés intellectuelles. Cela comprend le rappel ou la reconnaissance de faits précis, de modèles de procédures et de concepts qui servent le développement des capacités et compétences intellectuelles. Ce domaine contient six grandes catégories : connaissances, compréhension, application, analyse, synthèse/conception/création, évaluation.
- Le domaine affectif concerne les objectifs qui portent sur les émotions, les sentiments et les attitudes qui peuvent modifier les conditions d'apprentissage ou qui sont en rapport avec le domaine conceptuel du cours (Bilodeau et al., 1997 ; cité dans Bilodeau et al., 1999).
- Le domaine psychomoteur concerne l'acquisition d'une capacité psychomotrice (savoir-faire) ou d'un comportement (faire) (Bilodeau et al., 1997 ; Deschênes, 1991a ; cité dans Bilodeau et al., 1999).

2.3.1.2. Objectifs pédagogiques des TPs et télé-TPs

L'importance des travaux pratiques réside dans les objectifs pédagogiques *globaux particuliers* qu'ils permettent d'atteindre. Nous avons synthétisé dans le Tableau 2-1 ces objectifs à partir de la littérature étudiée : (Lelevé, 2002 ; Cooper, 2002 ; Hofstein et Lunetta, 2003 ; Millar, 2004 ; Singer et al., 2005 ; Benmohamed, 2006 ; Ma et Nickerson, 2006 ; ABET, 2009a ; ABET, 2009b ; ABET, 2009c ; ABET, 2009d.

Une première lecture du tableau montre que ces objectifs couvrent les trois domaines de la taxonomie de Bloom, et révèle la richesse et la complexité des activités de TPs qui peuvent être proposées.

Tableau 2-1. Objectifs pédagogiques des travaux pratiques.

Objectif		Description
Compréhension conceptuelle (Ma et Nickerson, 2006)		Améliorer la compréhension des faits et concepts, ainsi que la manière dont ils sont organisés dans les disciplines scientifiques.
Améliorer la maîtrise des objets, concepts et faits scientifiques (Singer et al., 2005)		Etablir le lien entre le domaine des objets et le domaine des idées (Millar, 2004).
Développer des compétences de conception et d'investigation (Ma et Nickerson, 2006).	Compréhension de la nature de la science (Singer et al., 2005)	Promouvoir la capacité de l'apprenant à identifier et à conduire des investigations scientifiques, ainsi qu'à formuler et défendre une argumentation scientifique.
	Développement du raisonnement scientifique (Singer et al., 2005)	Aider les apprenants à comprendre les valeurs et les hypothèses inhérentes à l'élaboration et l'interprétation des connaissances scientifiques.
Développer les compétences et habilités pratiques (Singer et al., 2005)		Apprendre à utiliser le matériel scientifique et les conventions de la science.
Compétences professionnelles (Ma et Nickerson, 2006)		Développer des compétences techniques et procédurales, introduire les apprenants dans le monde des scientifiques et ingénieurs, appliquer les connaissances dans la pratique.
Développer le travail et l'esprit d'équipe (Singer et al., 2005)		Favoriser la capacité d'un étudiant à collaborer efficacement avec les autres.
Développer des compétences sociales (Ma et Nickerson, 2006)		Acquérir des compétences sociales et autres comportements productifs (communication, résolution de problème, etc.).
Cultiver l'intérêt pour l'apprentissage des sciences (Singer et al., 2005)		Inciter les étudiants à apprendre davantage la science, et la considérer pertinente pour la vie quotidienne.
Comprendre la complexité et l'ambiguïté de travaux empiriques (Singer et al., 2005)		Aider les élèves à comprendre concrètement la complexité et l'ambiguïté des phénomènes naturels, mais aussi à relever les défis inhérents à l'observation et à la manipulation directe du monde matériel.

En ce qui concerne les télé-TPs, une étude menée par (Ma et Nickerson, 2006) a abouti aux résultats suivants :

- Les télé-TPs existants se focalisent sur les objectifs de compréhension conceptuelle, les compétences professionnelles et les compétences de conception, mais peu ou pas sur les compétences sociales.
- Les télé-TPs fondés sur les laboratoires virtuels s'intéressent davantage aux objectifs de compréhension conceptuelle et de compétences professionnelles qu'aux compétences de conception.
- Les télé-TPs fondés sur les laboratoires distants se focalisent sur les objectifs de compréhension conceptuelle et sur les compétences professionnelles.

L'étude a conclu que les compétences sociales sont négligées dans la plupart des travaux analysés alors qu'elles constituent une classe d'objectifs très importante. Toutefois, ces résultats ne peuvent être généralisés car ils concernent des télé-TPs intégrés dans des cursus traditionnels, en complément à d'autres activités d'apprentissage en présence. Le télé-TP lui-même est préparé dans la classe et parfois précédé d'une préparation dans un laboratoire réel, ce qui explique que des objectifs relatifs aux compétences conceptuelle et sociale ne sont pas visés. Pourtant, des travaux comme ceux de (Pernin, 1996 ; Buitrago, 1999 ; Guerraud, 2005 ; Benmohamed, 2006 ; Gravier, 2007) ont démontré qu'il est possible de couvrir tous les objectifs pédagogiques des TP via les télé-TPs lorsque ces derniers sont intégrés dans une démarche globale de conception pédagogique reposant sur les théories d'apprentissage, et qu'ils exploitent des plateformes et outils de formation à distance.

2.3.2. La conception pédagogique

(Singer et al., 2005) insistent pour que les travaux pratiques soient d'une part élaborés selon un processus de conception pédagogique (*Instructional Design*), et d'autre part intégrés dans le processus de conception global de tout le cursus de formation au côté des autres activités d'apprentissage.

2.3.2.1. Introduction à la conception pédagogique

La conception pédagogique englobe alors l'ensemble des processus d'analyse des besoins et de définition des objectifs d'apprentissage, ainsi que l'élaboration d'un système de diffusion pour

répondre à ces besoins. La conception pédagogique comprend l'élaboration du matériel didactique et des activités, mais aussi la mise au point et l'évaluation (Berger et Kam, 1996). Elle repose sur une ou plusieurs théories d'apprentissage qui essaient d'expliquer comment l'apprentissage prend place chez l'apprenant. Ainsi, le concepteur peut suivre différentes méthodes pour concevoir une activité d'apprentissage selon deux grandes catégories (Azland, 2010) : les méthodes informelles qui consistent à utiliser des recommandations plus ou moins précises, et les méthodes formelles qui définissent des procédures et étapes précises.

2.3.2.2. Les théories d'apprentissage

Les théories d'apprentissage tentent de décrire la façon dont les humains apprennent, ce qui nous aide à comprendre les processus complexes inhérents de l'apprentissage. Elles désignent « un ensemble de lois ou de principes qui décrivent la manière dont l'apprentissage se déroule » (Depover et al., 2006a). Selon ce dernier auteur, les concepteurs de formations, en plus d'être spécialistes du domaine à enseigner, doivent posséder un minimum de connaissances sur ces théories afin de concevoir des cursus capables de favoriser et susciter l'apprentissage chez l'apprenant. Les théories d'apprentissage sont souvent liées et confondues aux modèles d'apprentissage ; dans la suite du document, nous emploierons indifféremment ces deux termes. L'objectif ici n'est pas d'entrer dans les détails de ces modèles, nous renvoyons le lecteur intéressé à ces ressources (Mergel, 1998 ; Depover et al., 2006a) ; nous nous contentons d'en dresser un aperçu.

Historiquement, il existe trois grandes familles de modèles d'apprentissage apparues dans l'ordre chronologique suivant (Depover et al., 2006a) :

- Le *Behaviourisme* : ce modèle s'intéresse à l'étude des comportements et des changements observables. Par conséquent, il met l'accent sur la création de nouveaux comportements.
- Le *Cognitivisme* : il s'intéresse à examiner les processus de pensée inhérents au comportement. Il souligne l'acquisition et la réorganisation des structures cognitives.
- Le *Constructivisme* : cette théorie affirme que les connaissances se construisent à travers l'interaction entre les connaissances existantes et les expériences individuelles ou sociales.

Chacune de ces théories a évolué et continue d'évoluer pour donner de nouvelles théories qui essaient de combler certaines lacunes, tout en s'adaptant aux nouvelles formes de formation, notamment avec l'émergence de l'e-formation.

2.3.2.3. Les théories d'apprentissage pour les TP et télé-TPs

D'après les critères identifiés par (Mergel, 1998) pour choisir un modèle d'apprentissage (opérationnalisation de l'apprentissage, facteurs d'influence, rôle de la mémoire, transfert du savoir, et pratiques d'apprentissage visées), la théorie socioconstructiviste et ses variantes semblent être celles qui expliquent et mettent en œuvre le mieux l'apprentissage durant une activité de travaux pratiques (Lelevé, 2002). D'une part cette théorie est dominante dans l'enseignement des disciplines STEM (*Sciences, Technology, Engineering and Mathematics*) qui intègrent parmi leurs activités les TP, et d'autre part une activité de travaux pratiques prend souvent place dans un environnement social (le laboratoire) où des interactions riches s'établissent entre apprenants, tuteurs et environnement.

En particulier, la théorie privilégiée pour l'apprentissage des sciences et de l'ingénierie dans le cadre des TP est le modèle d'apprentissage par investigation (*Inquiry Based Model*) (Hofstein et Lunetta, 2003 ; Millar, 2004 ; Singer et al., 2005 ; Rocard et al., 2007 ; SCORE, 2008 ; ABET 2009a,b,c,d). En effet, l'apprentissage par investigation encourage et favorise l'apprentissage de la science en faisant la science (Aubé et David, 2003), et vise un double objectif : favoriser l'apprentissage via la méthode scientifique d'investigation, et apprendre à maîtriser cette méthode scientifique (Van Joolingen et Zacharia, 2009). Par ailleurs, cette méthode contribue à atteindre l'objectif général de développement du raisonnement scientifique chez l'apprenant (Millar 2004 ; Singer et al., 2005). L'apprentissage par investigation est souvent décrit comme un cycle (voir Figure 2-4) où l'apprenant est amené à formuler des questions pertinentes, prédire des réponses (en posant des hypothèses), mener une investigation (par le recueil de données, l'analyse des données, l'expérimentation, etc.), rédiger un compte rendu, et enfin mener une discussion et une réflexion fondées sur les résultats (Bishop et al., 2004). Ainsi, au lieu de transmettre des faits et concepts scientifiques considérés comme valides à l'apprenant, ce dernier apprend lui-même à construire ses connaissances par leurs confrontations aux données du monde réel tout comme le font les scientifiques et les professionnels.

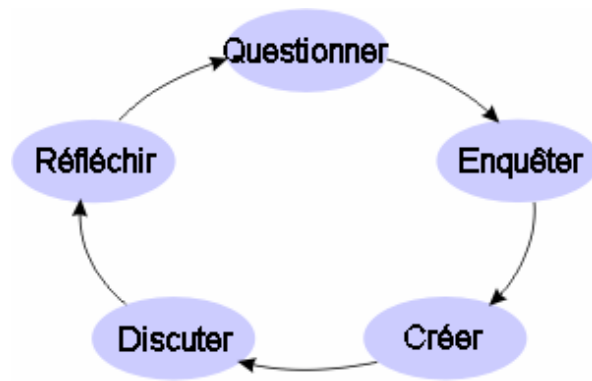


Figure 2-4. Le cycle du modèle d'apprentissage par investigation.

Toutefois, ce mode d'apprentissage suppose que les apprenants possèdent des aptitudes, des compétences et des connaissances approfondies dans le domaine, alors qu'en réalité ils ne les ont pas encore acquises. (Millar, 2004) souligne alors que dans la pratique, cette approche souffre de beaucoup de difficultés qu'il classifie en trois types :

- L'inexpérience de l'apprenant, la qualité des équipements utilisés et la durée dédiée aux travaux pratiques font que les apprenants formulent des observations et prennent des mesures qui ne sont pas suffisamment précises pour atteindre les objectifs visés.
- Même lorsque les apprenants obtiennent des données satisfaisantes en quantité et qualité, ils ne sont pas capables d'en faire une explication scientifique s'ils ne disposent pas des connaissances théoriques sous-jacentes.
- L'apprenant sait pertinemment que l'enseignant connaît la bonne réponse et qu'au définitif elle va lui être transmise. Le TP devient alors une sorte de jeu manipulant un réservoir d'informations scientifiques « prouvées et valides », et non pas une authentique situation de découverte de nouvelles connaissances.

Cette méthode pour l'enseignement de la science a également été fortement critiquée par (Kirschner, 1992), qui insiste sur le fait que l'expérience et la théorie sont interdépendantes : l'expérience construit la théorie, et la théorie détermine et encadre les expériences à réaliser. D'où la nécessité, selon l'auteur, d'une solide préparation théorique et de compétences métacognitives formelles pour que l'investigation soit efficace comme moyen de formation. (Van Joolingen et Zacharia, 2009) ont alors identifié quatre éléments satisfaisant les besoins exprimés par (Kirschner, 1992) et favorisant l'apprentissage par investigation :

- La définition de la mission de l'investigation : définir le scénario de l'investigation, assorti des objectifs pédagogiques et des buts visés afin de motiver et engager l'apprenant.

- La richesse et la pertinence des sources d'information : les ressources d'apprentissage traditionnelles, les simulations et les expériences en laboratoires.
- L'utilisation d'outils adéquats d'expression des connaissances : les outils de communication des résultats de l'apprentissage sous forme de comptes-rendus, modèles, graphes, cartes conceptuelles, etc.
- Les étayages sociaux et cognitifs : il s'agit ici des ressources, outils, personnes ainsi que de l'environnement à mettre à disposition des apprenants afin qu'ils acquièrent les compétences leur permettant de mener une investigation. (Hmelo-Silver et al., 2006) soulignent que le guidage doit être assuré par un enseignant ou un expert humain ou informatique qui fournit les explications nécessaires au moment opportun.

2.3.3. Les activités d'apprentissage

Une activité de travaux pratiques est conçue en fonction du domaine enseigné, du niveau des apprenants, du type et du mode d'enseignement, des objectifs pédagogiques généraux et spécifiques, ce qui donne lieu à une immense diversité des activités pratiques proposées. Toutefois, un rapport récent issu du groupe de travail SCORE en Grande Bretagne (SCORE, 2009) propose une classification des activités de travaux pratiques en trois grands groupes :

- Les activités pratiques principales (les investigations, les techniques et procédures en laboratoire, le travail sur le terrain).
- Les activités directement liées aux travaux pratiques (conception et planification des investigations, analyse des données en utilisant les technologies de l'information et de la communication, démonstrations par les enseignants, observation des phénomènes).
- Les activités complémentaires aux travaux pratiques (visites ou excursions scientifiques, enquêtes scientifiques, présentations et jeux de rôles, utilisation des simulations, modélisation, discussions en groupe, activités de rédaction en groupe).

La classification proposée ci-dessus démontre qu'un (télé-)TP est plus qu'une suite d'expérimentations isolées. Cette activité doit intégrer des tâches diversifiées de lecture, d'exercices, de réflexion, de travail collaboratif, de rédaction et de communication. Mais elle doit aussi s'insérer dans une séquence de formation et interagir avec les autres activités d'apprentissage afin de participer à l'atteinte des objectifs pédagogiques de cette séquence.

2.3.4. La présence

Selon (Sheridan, 1992 ; cité dans Ma et Nickerson, 2006), il existe trois types de présence :

- La présence physique : être physiquement présent dans un lieu donné.
- La télé-présence : le sentiment d'être dans un endroit éloigné du lieu réel de l'opération.
- La présence virtuelle : le sentiment d'être présent dans un environnement virtuel généré par l'ordinateur.

Des chercheurs et éducateurs affirment que la présence physique dans le laboratoire est un facteur important pour la réussite d'un TP, et avancent les effets positifs de la « *physicalité*⁷ » sur les performances de l'apprentissage à travers les manipulations physiques de dispositifs et instruments de laboratoires (Zacharia et Olympiou, 2010). La *physicalité* est définie comme le toucher actif et réel d'un objet physique (Zacharia et Olympiou, 2010). Ils affirment que rien ne peut remplacer ce type de manipulation, et que la télé-présence et la présence virtuelle ne peuvent représenter une alternative. (Kirschner, 1992) affirment que les travaux pratiques en laboratoire ont une fonction psychologique de familiarisation avec le monde réel, puisqu'ils procurent du « plaisir » aux apprenants qui « sentent » l'expérience et construisent des connaissances tacites sur ces expériences. C'est l'occasion pour les apprenants d'acquérir non pas des compétences formelles, mais des compétences et connaissances ayant une des connotations verbales, musculaires, émotionnelles et intellectuelles qui s'intègrent dans le domaine des compétences affectives et psychomotrices.

Ce point de vue est confirmé par la plupart des sondages effectués sur la perception par les apprenants des télé-TPs qui démontrent que la majorité d'entre eux préfère les expériences réelles en laboratoires (Ma et Nickerson, 2006) ; y compris la catégorie des handicapés qui pensent que les télé-TPs vont encore les exclure et les éloigner de l'environnement d'apprentissage traditionnel (Colwel, 2002).

Toutefois, l'étude de (Zacharia et Olympiou, 2010) a conclu que même si les manipulations mobilisant le maximum de modalités sensorielles donnent de bons résultats d'apprentissage, elles ne sont pas plus efficaces que les manipulations virtuelles, sauf lorsque l'objectif principal est de maîtriser des habilités motrices. De plus, la présence physique en laboratoire

⁷ Traduction du mot anglais « physicality »

peut être un handicap à l'apprentissage. Par exemple, lorsqu'un grand nombre d'apprenants sont présents, tous n'ont pas l'occasion de réaliser l'expérience. Pour les handicapés moteurs, le problème est encore plus compliqué puisqu'ils sont souvent privés d'expérimentation (Colwel, 2002). Par ailleurs, certains objets et phénomènes physiques ne peuvent être perçus par les cinq sens de l'Humain, seuls les environnements informatiques en illustrent la forme et le comportement (Zacharia et Olympiou, 2010).

La présence psychologique et sociale est le sentiment d'être ensemble, le sentiment d'interactions sociales dans des espaces partagés avec des personnes situées dans un autre endroit qui peut être distant (Witmer et Singer, 1998 ; cité dans Müller et Erb, 2007). La présence sociale est donc un facteur important pour assurer une communication et une collaboration efficaces, malheureusement difficiles à reproduire complètement dans un environnement d'apprentissage distribué. Toutefois, des études ont démontré que si l'environnement informatique utilise convenablement les variables psychologiques et sociales de la présence, ce sentiment peut être reproduit et affecter positivement les performances de l'apprentissage pourvu que l'environnement informatique soit convaincant ; les travaux de (Ma et Nickerson, 2006 ; Müller et Erb, 2007) suggèrent par exemple d'utiliser les techniques de la réalité mixte comme solution au renforcement de la présence sociale. Parmi ces variables psychologiques qui peuvent agir sur le sentiment de présence, citons :

- Le degré de réalisme des expériences ou le degré de fidélité à la réalité.
- La représentation virtuelle du lieu, fondée sur la métaphore spatiale.
- Les sentiments de présence et d'appartenance, reposant sur l'*Awareness* ou la conscience collective (mutuelle).

La section suivante s'intéresse au degré de réalisme des expériences, alors que les deux dernières variables sont traitées dans les sections consacrées à l'apprentissage collaboratif et au tutorat durant un télé-TP.

2.3.5. La fidélité de la télé-instrumentation

La télé-instrumentation de laboratoires distants ne s'applique évidemment qu'aux télé-TPs, et constitue l'aspect dont les projets relatifs à ces activités se focalisent le plus (Ma et Nickerson, 2006). La télé-instrumentation comprend l'ensemble des mécanismes, protocoles et IHM disponibles aux utilisateurs finaux pour interagir avec le laboratoire distant. Les travaux

existants se concentrent sur les télé-opérations synchrones (BenMohammed, 2006 ; Gravier, 2007), mais les télé-TPs offrent une alternative intéressante à travers la télé-instrumentation asynchrone. En particulier lorsqu'il s'agit de tâches longues et fastidieuses de mise en marche de processus de mesures d'un phénomène, de traitement d'un grand flux de données par un programme de calcul, etc.

La fidélité définit le degré de réalisme de la simulation ou de l'IHM du laboratoire distant qui instrumente la télé-opération. Elle se distingue par la fidélité physique ou technique qui se concentre sur la similitude exacte de l'environnement informatique avec le laboratoire ou l'expérience physique, et la fidélité psychologique qui agit sur la perception, par l'individu, du réalisme de l'expérience ou du laboratoire.

Des études ont démontré que la fidélité psychologique peut être aussi forte que la fidélité physique (Ma et Nickerson, 2006). A l'inverse, d'autres travaux considèrent qu'une fidélité physique élevée va d'une part aider l'apprenant à reconnaître l'objet ou le phénomène lorsqu'il le rencontre dans la réalité, et d'autre part fournir des étayages de perception pour les concepts abstraits difficiles à démontrer autrement. Enfin, certains affirment que moins l'environnement est réaliste, plus l'apprentissage est efficace, car l'élimination des détails inutiles diminue la charge cognitive de l'apprenant et l'aide à se concentrer sur l'objet du TP (Zacharia et Olympiou, 2010).

La fidélité soutient donc les processus d'inférence et de raisonnement fondés sur la perception. Des recherches ont démontré que la technologie et le lieu du laboratoire n'ont pas un impact important sur l'apprentissage, contrairement au degré de motivation, à la qualité de la collaboration et des *feedbacks*, et à la richesse des médias utilisés (Ma et Nickerson, 2006). Enfin, le réalisme est important pour les activités dont l'objectif principal est l'acquisition de compétences professionnelles, où l'apprenant doit maîtriser dans la réalité les dispositifs technologiques et industriels qu'il sera amené à utiliser après sa formation.

2.3.6. L'apprentissage collaboratif

Comme nous l'avons énoncé, le travail en groupe constitue un objectif pédagogique à part entière pour les activités de travaux pratiques (Singer et al., 2005 ; Hofstein et Lunetta, 2003 ; Millar, 2004). L'apprenant, futur scientifique ou ingénieur, sera amené à travailler en groupe aussi bien dans le domaine de la recherche que dans le monde de l'ingénierie pour trouver des

solutions à des problèmes complexes. L'acquisition de cette compétence nécessite d'engager les apprenants dès le début de leurs formations dans des activités collectives d'apprentissage. Par ailleurs et dans le contexte de l'apprentissage par investigation, la collaboration est fondamentale dans les processus de construction de nouvelles connaissances et de correction des préconceptions erronées existantes chez les apprenants.

(Hofstein et Lunetta, 2003) soulignent que les travaux pratiques en laboratoire dans l'enseignement traditionnel constituent l'unique environnement d'apprentissage collaboratif engageant des groupes restreints dans des activités d'investigation scientifique. Les auteurs considèrent que les TP ont le potentiel d'activer des relations sociales collaboratives ainsi que des attitudes positives envers la croissance de la cognition et la science. Des études empiriques ont confirmé que ce mode d'apprentissage permet d'atteindre de meilleures performances, des résultats scolaires supérieurs, une motivation accrue, un développement important de la pensée critique et une plus grande satisfaction des étudiants (Karsenti et Fortin, 2003).

En situation de formation à distance, l'apprentissage collaboratif est supporté par des systèmes dédiés (CSCL⁸, CSCW⁹) où les interactions et communications en face-à-face sont remplacées par deux types d'outils :

- Les outils de communication asynchrones, comme le courrier électronique et les forums de discussions, qui offrent une communication souple et flexible dans le temps et un enregistrement des communications.
- Les outils de communication et de collaboration synchrones comme la messagerie instantanée, l'audio/visioconférence, le tableau blanc ou les applications partagées qui se sont révélés utiles pour maintenir la motivation. Toutefois, bien qu'ils restent indispensables, l'environnement d'apprentissage doit intégrer des indicateurs, des repères, tableaux de bord et autres outils renforçant le sentiment de présence virtuelle et de présence sociale.

Différents sondages affirment que les apprenants préfèrent le travail d'équipe en présentiel car les interactions sont plus riches et diversifiées (orales, gestuelles, etc.), mais des études

⁸ Computer Supported Collaborative Learning

⁹ Computer Supported Cooperative Work

montrent que les résultats d'apprentissage obtenus en situation présentielle peuvent être atteints dans le cadre de l'apprentissage collaboratif à distance (Karsenti et Fortin, 2003).

Par ailleurs, (Ma et Nickerson, 2006) soulignent que la complexité des problèmes de recherche et d'ingénierie nécessite des ressources humaines et des moyens scientifiques et techniques plus importants que ceux disponibles dans un seul laboratoire de recherche. Des réflexions permettant de mutualiser les ressources humaines et matérielles des laboratoires et organismes de recherche afin de résoudre ces problèmes ont été menées (Scanlon et al., 2004). La collaboration médiatisée par ordinateur (Agarwal et Johnston, 1998), la télé-opération de dispositifs distants (Brian et al., 2002 ; Fiuczynski, 2006 ; Owezarski et al., 2008), et l'exploitation de ressources partagées de traitement, d'analyse et de stockage de données (Jones, 2005) ont été retenues comme solutions pertinentes à ce problème. Elles éliminent les barrières de communication et de collaboration entre chercheurs, et étendent les expériences aux delà des moyens d'un laboratoire local.

2.3.7. Le tutorat

Le rôle de l'enseignant est critique dans les travaux pratiques puisque ce dernier est chargé d'introduire des expériences assurant l'atteinte des objectifs d'apprentissage, la bonne orientation des discussions et réponses aux questions, ainsi que l'établissement de liens entre les concepts de l'activité et les concepts théoriques sous-jacents. L'enseignant ou le tuteur doivent également évaluer, de manière formative, les apprenants tout au long de l'activité. Cette évaluation va permettre de mesurer l'état d'avancement des apprenants et recenser les problèmes rencontrés.

Dans le cadre des TPs traditionnels, cette tâche délicate est généralement effectuée par l'observation des comportements des apprenants durant la réalisation d'opérations spécifiques ou pendant l'activité toute entière, au moyen d'évaluations des productions orales et écrites, d'analyses des discussions, etc. (Singer et al., 2005). L'évaluation formative interagit avec la fonction de tutorat en lui fournissant des informations pertinentes permettant à l'enseignant d'intervenir de façon adéquate au moment opportun.

Dans le domaine des télé-TPs, le tutorat est identifié (aussi bien par les théories que par les acteurs) comme une fonction pédagogique très importante qui doit être supportée, et il est considéré par la théorie socioconstructiviste comme une collaboration entre des apprenants et

un tuteur qui joue le rôle d'expert, de facilitateur, de *coach* (Dabbagh, 2005). Le problème qui se pose pour le tutorat des télé-TPs est la visibilité des activités des apprenants que ce soit dans le cadre d'activités individuelles ou collaboratives. Dans ce contexte, deux types de tutorat doivent être distingués : le tutorat synchrone et le tutorat asynchrone. Le tutorat asynchrone utilise les outils de communication asynchrone décrits plus haut pour venir en aide aux apprenants, mais également les traces d'apprentissage (sous forme de documents déposés, d'historique des communications, d'historique des accès aux ressources) pour évaluer l'apprentissage. Mais ce type de tutorat n'est pas suffisant puisque l'apprenant a souvent besoin d'une aide précise et ponctuelle que peut lui apporter le tuteur ou un autre apprenant, d'une réponse à une question, de voir et savoir ce que les autres font pour coordonner les actions comme c'est le cas lors d'un TP traditionnel. Le modèle de Suivi d'Activités d'Apprentissage à Distance SAAD (Després et Leroux, 2003), amélioré par les extensions de (Guerraud, 2004), s'attache à répondre à ces besoins. Ce modèle est issu de travaux en relation avec la robotique pédagogique et le projet ESSAIM (Després, 2001). Celui-ci a été étendu et implémenté une autre fois dans le cadre du projet FORMID, un environnement d'apprentissage support aux télé-TPs utilisant les simulations pédagogiques. Les détails de ces modèles et leurs implémentations seront présentés dans le prochain chapitre.

L'apprentissage collaboratif et l'activité de tutorat sont des notions complexes à mettre en œuvre dans le cadre des télé-TPs, puisqu'elles font appel à deux concepts : les outils de renforcement de l'*Awareness* et la notion d'espaces de travail partagés. Ces derniers points, souvent négligés dans les projets de télé-TPs (Jara et al., 2009 ; Tee et al., 2006), font l'objet de la section qui suit.

2.3.8. L'*Awareness*

Nous reprenons ici en détail quelques éléments que nous estimons importants pour favoriser un apprentissage collaboratif efficace dans le contexte des télé-TPs.

2.3.8.1. L'*Awareness* ou la conscience mutuelle (collective)

(Dourish et Belloti, 1992) définissent l'*Awareness* comme « la compréhension des activités des autres, qui fournit un contexte pour notre propre activité » - [...] - « ce contexte est utilisé

pour assurer que les contributions individuelles sont pertinentes à l'activité du groupe tout entier et évaluer les actions individuelles en respectant les buts et l'évolution du groupe ».

Pour (Faerber, 2003), ce terme désigne « la perception que possède chacun de (1) la présence, (2) la localisation, (3) l'identité, (4) la disponibilité de l'autre à un moment donné », ainsi que « la perception de ce qui a été réalisé entre deux connexions successives et de l'historique de l'activité du groupe ». Cette dernière propriété est désignée par (Temperman et al., 2007) par l'*Awareness* asynchrone, alors que les quatre premières désignent l'*Awareness* synchrone.

Diverses classifications de l'*Awareness* ont été identifiées et se chevauchent, ce qui témoigne de la complexité de ce concept. Elles ont généré différents modèles conceptuels guidant le processus de conception d'IHMs pour les environnements de collaboration (Gross et Wolfgang, 2003 ; Markopoulos et De Ruyter, 2009 ; Rittenbruch et McEwan, 2009), mais nous nous focaliserons dans le cadre de cette thèse uniquement aux types d'*Awareness* qui permettent de supporter les expérimentations collaboratives : l'*Awareness* des espaces de travail, et l'*Awareness* des artefacts. Notons que le concept d'*Awareness* doit être conçu et implémenté avec prudence en ce qui concerne les questions de sécurité et d'intimité, au risque d'être rejeté par les utilisateurs (Min-Kyung et Hee-Cheol, 2007 ; Patil et Kobsa, 2009).

2.3.8.2. L'*Awareness* des espaces de travail

La mise en œuvre de l'apprentissage collaboratif nécessite des espaces communs de travail collaboratif (Faerber, 2001c ; Dillenbourg et al., 2002 ; Henri et Basque, 2003 ; Schaf et al., 2009). Un espace de travail collaboratif est un centre polyvalent donnant accès à certaines ressources et outils de communication ou de collaboration. Souvent une métaphore spatiale est utilisée pour structurer l'interface et donner des repères pour la navigation dans l'environnement (Henri et Basque, 2003 ; Faerber, 2001b). (Sonnenwald et al., 2004) suggèrent qu'un espace de travail commun fournisse trois types d'informations : les informations contextuelles, les informations sur les tâches et les processus, et les informations socio-émotionnelles. Ces derniers auteurs préconisent de faire appel aux techniques de la réalité virtuelle afin de maintenir le sentiment de présence parmi les acteurs, alors que d'autres études suggèrent les techniques de télé-présence et de présence virtuelle (Ma et Nickerson, 2006 ; Nickerson et al., 2007 ; Müller et Erb, 2007 ; Zacharia et Olympiou, 2010).

2.3.8.3. L'*Awareness* des artefacts

Les expérimentations collaboratives peuvent être supportées par l'*Awareness* des artefacts (Tee et al., 2006). Ces auteurs définissent ce concept comme la connaissance que chaque personne possède à un moment donné de l'utilisation individuelle des artefacts et outils par les autres acteurs dans le cadre de leurs travaux individuels. L'expérience étant présentée à l'utilisateur comme un artefact informatique partagé, apprenants et tuteurs doivent savoir ce que chacun fait sur cette expérience afin d'agir en conséquence. Si les outils de communication sont les plus utilisés pour supporter ces interactions, dans le cadre d'expérimentations collaboratives où les acteurs télé-opèrent des dispositifs physiques ou virtuels, il n'existe pas d'outil standard pour voir ces actions de télé-opérations : les moyens de télésurveillance utilisés dans certains projets de télé-TPs présentés dans le prochain chapitre (caméras, microphones ou autres types de sondes et capteurs) rendent compte uniquement de l'état du laboratoire en cours d'utilisation mais n'associent pas d'une manière exacte les télé-opérations sur un dispositif donné à l'acteur humain qui les a exécutées. Des solutions sont proposées dans le cadre des simulations où l'état de la simulation est synchronisé entre les différents acteurs (Jara et al., 2009), ou par le partage partiel ou total d'écrans de travail (Tee et al., 2006).

Nous avons identifié dans cette section, à partir d'une revue de la littérature, huit dimensions pour l'analyse de l'efficacité des activités de télé-TPs qui sont interdépendantes. En effet, la définition des objectifs pédagogiques influe sur le choix de la théorie et du design pédagogique, alors que ces deux derniers influent sur le choix du type d'activité d'apprentissage (exercice, projet, etc.). La majorité des références que nous avons utilisées pour construire ce modèle d'analyse considère les télé-TPs comme compléments aux enseignements traditionnels dispensés en présence ; or, de plus en plus de formations sont dispensées partiellement ou complètement à distance, et il nous paraît évident que les télé-TPs doivent être intégrés aux EIAHs existants pour pouvoir être diffusés à grande échelle et acceptés par le plus grand nombre d'acteurs de la formation. Dans la prochaine section, nous discuterons de cette intégration ainsi que des contraintes et problèmes liés à sa réussite.

2.4. Intégration des télé-TPs en EIAH

Nous avons vu que la plupart des recommandations pour atteindre l'efficacité pédagogique des travaux pratiques insiste sur l'intégration de ces derniers dans la démarche globale de conception pédagogique de tout le cursus de la formation scientifique ou d'ingénierie. Aussi, les activités de travaux pratiques elles-mêmes doivent être conçues selon cette même démarche (Singer et al., 2005).

Ces recommandations dans le cadre des formations à distance consistent à intégrer les travaux pratiques en ligne dans le flux des activités d'apprentissage délivrées par l'EIAH global de toute la formation (Lelevé et al., 2003; Benmohamed, 2006 ; Gravier, 2007). A cet égard, les télé-TPs doivent être conçus en assurant leur interopérabilité et réutilisabilité vis-à-vis d'EIAH hétérogènes (Lelevé et al., 2004). Afin de réussir cette intégration, nous pensons qu'il faut prendre en compte les progrès réalisés dans le domaine des EIAH en ce qui concerne la gestion et l'organisation de cursus en ligne, la gestion et l'exploitation des ressources d'apprentissage, ainsi que les outils supports aux différentes fonctions pédagogiques (conception, apprentissage, tutorat).

2.4.1. Introduction aux EIAH

Un EIAH est un environnement informatique conçu dans le but de favoriser l'apprentissage humain, c'est-à-dire la construction de connaissances chez un apprenant (Tchounikine, 2002). En tant que champ scientifique, il contient la synthèse de l'évolution des environnements dédiés à la formation à distance. (Murray, 1999 ; cité dans Tchounikine, 2002) distingue deux grandes classes d'environnements d'apprentissage : les environnements orientés pédagogie, et les environnements orientés performance.

Les environnements orientés pédagogie trouvent leurs origines dans la formation à distance qui a commencé au milieu du XIX siècle en Grande Bretagne à travers les cours par correspondance évoluant jusqu'aux environnements de formation (ouverte et) à distance médiatisés par les technologies (Peraya, 2001). Ces environnements sont centrés sur l'organisation de l'apprentissage, la diffusion de ressources numériques et la mise à disposition d'outils de communication. Les plateformes de e-Learning (ou Learning Management Systems, LMS) ainsi que les plateformes spécialisées dans la production, la

médiatisation et l'organisation des ressources d'apprentissage (ou Learning Content Management Systems, LCMS), représentent de tels environnements. L'apprentissage via une plateforme de FOAD est lié d'une part à la conception pédagogique des activités d'apprentissage qui sont intimement liées aux théories d'apprentissage, et d'autre part au design pédagogique.

Les environnements orientés performance « visent à créer une situation d'apprentissage, i.e. une activité individuelle ou collective conçue pour favoriser l'apprentissage de concepts précis et/ou l'acquisition de compétences précises, et pour lesquelles les interactions (apprenant - apprenant, apprenant - tuteur, apprenant - système) et les processus d'apprentissage dont elles sont le cadre sont considérés comme des facteurs clés, au centre de l'étude. » (Tchounikine, 2009). Les premiers environnements de ce type remontent aux années soixante avec les systèmes d'enseignement assistés par ordinateur (EAO), qui ont évolué vers des systèmes intégrant les principes de l'intelligence artificielle appelés systèmes de tutorat intelligent (ou tuteurs intelligents). Ces derniers ont muté vers les Environnements Intelligemment Assistés par Ordinateurs (EIAO) et les Micro-Mondes. Les EIAO sont ensuite devenus des systèmes portant le même nom mais avec un sens différent : Environnement Interactif Assisté par Ordinateur centré sur l'interactivité de l'apprenant avec le système comme facteur favorisant l'apprentissage (Bruillard et al., 2000). Cette évolution découle de celle des médias et des technologies, mais aussi des théories d'apprentissage. Les premiers systèmes, fondés sur le *behaviourisme* puis le *cognitivism*, appliquaient les principes de l'enseignement programmé qui consistent à « transférer » les connaissances de l'ordinateur vers l'apprenant. L'avènement du *constructivism* après la remise en cause de ces théories, notamment par les travaux de Piaget, a conduit aux EIAOs fondés sur l'interaction entre apprenant et ordinateur pour la construction active de nouvelles connaissances, et aux Micro-Mondes (Papert, 1981). Mais l'évolution la plus récente de ces théories a mené à la création du champ scientifique de recherche en EIAH, un domaine transdisciplinaire alliant informatique, technologies et sciences humaines et sociales pour développer des environnements capables de susciter l'apprentissage chez l'apprenant. Des normes et standards pour la description, l'indexation, l'intégration et l'exécution de ressources, ainsi que des méta-modèles de scénarisation pédagogique représentent les fondations de ces environnements informatiques.

2.4.2. Les standards pédagogiques

Les normes et standards pédagogiques ont émergé suite à la quantité importante de ressources numériques hétérogènes nécessitant beaucoup d'investissements pour leur adaptation dans des contextes différents. La normalisation des ressources d'apprentissage s'est transformée en un enjeu pédagogique, culturel et économique majeur ; pour preuve la multitude d'organismes et de groupes impliqués dans cet effort (Arnaud, 2002). L'intérêt de ces standards pédagogiques est d'assurer l'accessibilité, l'interopérabilité, la réutilisabilité, la durabilité et l'adaptabilité des ressources, facilitant ainsi leur partage entre enseignants indépendamment de toute plateforme (SCTIC, 2002). Les normes peuvent être classées selon différentes perspectives et selon différents niveaux d'opérations (SCTIC, 2002) :

- Le niveau contenu s'intéresse à la description et à l'indexation d'unités de contenus permettant de les stocker dans les dépôts de ressources d'apprentissage en vue d'en faciliter la recherche, l'évaluation, l'intégration, l'adaptation et la recomposition. Ceci est réalisé par le renseignement de métadonnées suivant un standard comme le LOM (IEEE, 2009).
- Le niveau contrôle et exécution de contenus s'intéresse aux interfaces d'intégration, d'agrégation et d'exécution des contenus d'apprentissage dans les plateformes de formation. Le standard SCORM (ADLNet, 2009) est un exemple de standard assurant ces fonctions.
- Le niveau de modélisation pédagogique spécifie l'organisation dans le temps et dans l'espace d'une activité d'apprentissage en y explicitant tous les éléments. Les différents langages de modélisation pédagogique (*Education Modeling Language*, EML) en sont l'exemple, en particulier le langage IMS-LD (IMSGlobal, 2003).

Au regard du besoin pressant d'intégrer les télé-TPs dans les cursus de formation à distance, la question de leur normalisation à tous les niveaux se pose systématiquement (Lelevé, 2004 ; Benmohamed, 2006) ; il est évident que cela ne peut se faire qu'avec le respect des standards existants.

2.4.3. Positionnement des télé-TPs

Si l'on considère que l'interaction active avec des objets informatiques fait partie des télé-TPs, alors les premières formes de télé-TP seraient les Micro-Mondes (années 70) et les

simulations pédagogiques (années 90) qui peuvent être classées dans la catégorie des environnements orientés performance.

Au début du XXI siècle, les télé-TPs ont commencé à émerger dans les travaux de recherche pour combler un vide qui se ressent de plus en plus dans les formations (ouverte et) à distance en STEM (Lelevé, 2002 ; Gravier et al., 2008). Une diversité de projets traitant des télé-TPs dans divers domaines ont vu le jour, résultant souvent en des environnements et outils propriétaires, utilisables uniquement dans le contexte particulier du laboratoire ou de l'université à l'origine de l'outil, et exploitables via une IHM propriétaire intégrée ou non dans un EIAH. En fait, la plupart de ces projets ont considéré les télé-TPs comme une extension aux enseignements et aux travaux pratiques traditionnels, empêchant ainsi la capitalisation des expériences.

A partir de la revue de la littérature sur les différents projets de télé-TPs, C. Schmid, cité dans (Gustavo et al., 2007), a cependant constaté que les architectures des plateformes dédiées aux télé-TPs se ressemblent fortement. Il a alors proposé une architecture générique.

2.4.3.1. Du point de vue architecture

L'architecture générique élaborée par C. Schmid est constituée des éléments suivants illustrés par la Figure 2-5 :

- Un serveur d'expériences ou d'instrumentation : ce composant est chargé de médiatiser les dispositifs du laboratoire, et peut être démultiplié selon le dimensionnement du laboratoire distant. Pour diverses raisons, Schmid recommande que ce serveur soit caractérisé par des politiques d'accès, de réservation et de sécurité avancées.
- Un serveur de médias : ce composant médiatise les retours audio et vidéo des dispositifs d'expérimentation survenant au cours d'un télé-TP. Il est considéré comme les « yeux et oreilles » de l'utilisateur sur le laboratoire distant.
- Un serveur WEB : cet élément contient toutes les informations et ressources dont l'apprenant a besoin pour réaliser les expériences demandées. Il fournit également l'environnement d'apprentissage situé qui place l'expérience en ligne dans un fond théorique.

- Un serveur de contrôle d'accès : il contrôle l'accès aux ressources du laboratoire distant et prévient ainsi les personnes non autorisées à y accéder. Il peut également assurer la fonction de réservation des ressources.
- Un serveur de partage (fournisseur de ressources) : ce serveur est responsable du partage et de la mutualisation des ressources du laboratoire avec les institutions partenaires. En général, il implémente un Middleware offrant des services accessibles à ces institutions, permettant de construire une infrastructure de laboratoire virtuel.
- Le client utilisateur : le plus souvent c'est un client web embarqué dans le navigateur Internet. Son interface intègre l'IHM de télé-opération avec les autres ressources d'apprentissage (cours, exercices, guides, feuilles de route, etc.).

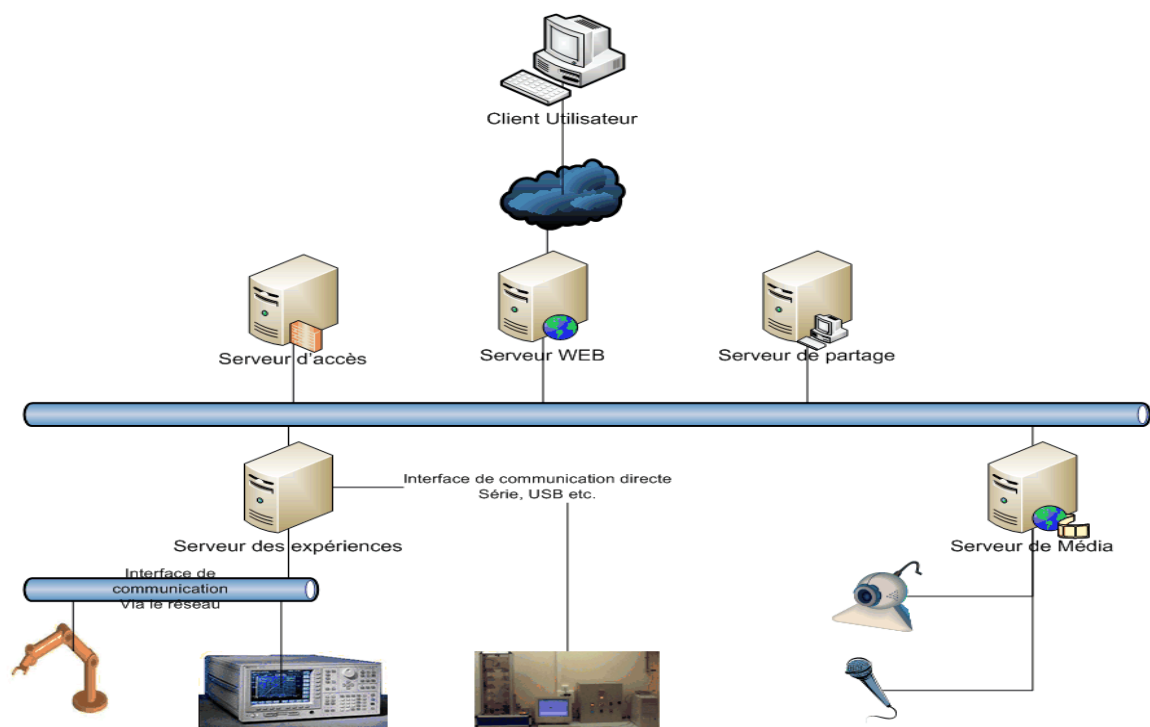


Figure 2-5. Architecture générale d'une plateforme de télé-TP (Gustavo et al., 2007).

Nous pensons que les télé-TPs doivent se placer à la croisée des deux catégories d'EIAH : tout en restant des environnements orientés performance car ils visent l'atteinte d'objectifs précis, ils doivent être intégrés dans des plateformes de FOAD pour qu'ils puissent être accessibles dans les cursus de formation en ligne. Ils bénéficieraient ainsi des avancées technologiques et pédagogiques en termes de design pédagogique, d'outils de conception d'objets d'apprentissage et d'outils supports à l'apprentissage collaboratif et au tutorat.

2.4.3.2. Du point de vue du respect des normes pédagogiques

Des recherches ont identifié deux solutions complémentaires pour la normalisation des télé-TPs d'une manière générale, et leur interopérabilité avec les standards existants en particulier (Buitrago, 1999 ; Gueraud, 2005 ; Benmohamed, 2006 ; Gravier, 2007). La première solution est un protocole de communication entre le laboratoire et l'EIAH (Pernin, 1996 ; Buitrago, 1999 ; Gueraud, 2005) issu de la recommandation du groupe de travail Communication Outil/Agent du comité IEEE-LTSA (IEEE, 2003). Ce comité suggère un protocole de communication entre l'outil d'instruction (l'expérience) et l'agent d'instruction (l'EIAH) qui doit :

- Permettre à l'agent d'instruction d'observer ce que l'apprenant réalise au sein des outils.
- Permettre à l'agent d'instruction de donner un *feedback* à l'apprenant.
- Transmettre à l'outil les demandes de l'agent d'instruction.
- Permettre aux agents d'instruction de communiquer entre eux pour partager leurs informations dans le but d'évaluer ou de guider l'apprenant.
- Établir un standard pour référencer les objets des outils.

Tout en garantissant l'indépendance du fonctionnement interne de chaque élément de cette architecture, les caractéristiques que le protocole cherche à implanter sur les outils et les agents sont :

- La capacité d'être observable, c'est-à-dire la capacité d'un outil ou agent d'informer automatiquement des événements survenant en son sein.
- La capacité d'être scriptable, ou modifiable, via un langage de commande.
- La capacité d'être inspectable pour être en mesure de délivrer une information précise suite à une demande explicite d'un autre outil ou agent.

Ce protocole ne suffit pas pour assurer la généricité des télé-TPs car son opérationnalisation nécessite la création de programmes de bas niveau qui communiquent avec le dispositif distant, ce qui est incompatible avec l'esprit de la standardisation qui vise justement à s'affranchir des spécificités physiques. La seconde solution consiste alors à utiliser un méta-modèle pour modéliser le laboratoire au niveau de la partie opérative et de la partie commande. Ainsi, intégrer un télé-TP revient à intégrer dans l'EIAH sa description, et le contrôler revient à lire les valeurs des variables du modèle et à invoquer leurs méthodes. Ces processus assurent l'échange et le partage de modèles au niveau conceptuel, et sont valables

pour différents laboratoires. Le modèle produit peut alors être décrit par le standard LOM de description de ressources, et même être exploité par un scénario pédagogique IMS-LD. Toutefois, il restera toujours une tâche de développement d'un programme établissant le lien entre le modèle descriptif de l'expérience et l'expérience elle-même.

2.5. Synthèse

Les travaux pratiques sont des enseignements indispensables dans les cursus scientifiques et d'ingénierie qui nécessitent d'importants efforts lors de leur préparation et réalisation, aussi bien de la part des apprenants que des enseignants. Pour atteindre l'efficacité pédagogique escomptée de ce genre d'activités, il faut les intégrer dans un processus de conception pédagogique qui prend en compte les théories d'apprentissage modernes. Les théories cognitiviste et constructiviste fournissent les modèles qui expliquent le mieux l'activité d'apprentissage dans le cadre de travaux pratiques, vu le caractère fortement interactif entre l'apprenant et l'environnement technique et social.

Le point de départ de ce processus est l'explicitation des objectifs pédagogiques globaux attendus ainsi que les objectifs spécifiques sous-jacents issus du domaine d'enseignement particulier. En fonction de ces objectifs et suivant le processus de conception pédagogique choisi, des ressources seront élaborées et/ou choisies par l'enseignant. Il n'existe pas de « recette » prête à l'emploi pour la conception d'une activité de travaux pratiques, mais un certain nombre de conditions doivent être réunies pour réussir une telle activité. Il s'agit essentiellement de l'apprentissage collaboratif qui favorise le travail d'équipe et l'apprentissage des pairs, l'expérimentation qui favorise l'acquisition de compétences pratiques et de connaissances tacites, le tutorat pour guider l'apprentissage et soutenir les apprenants en difficulté, et les moyens d'évaluation pour vérifier l'atteinte de l'efficacité pédagogique.

Les télé-TPs augmentent le niveau de complexité des TP traditionnels. Même si certaines avancées technologiques mûres offrent des environnements en ligne réalistes et fidèles aux environnements de laboratoires réels, des réticences quant à leur adoption persistent parmi un grand nombre d'enseignants et apprenants, face à l'enthousiasme d'une large communauté de formation qui y trouve une solution pertinente pour faire des économies et couvrir les besoins d'une formation de qualité à un nombre de plus en plus important d'étudiants. Toutefois, la

plupart des chercheurs et enseignants s'accorde sur l'efficacité des télé-TPs : elle doit être au moins égale à celle des TP pour que les activités de télé-TPs soient éligibles à leur intégration dans un cursus de formation scientifique partiellement ou totalement en ligne. La mesure de cette efficacité dans le cadre des télé-TPs s'effectue de la même manière que pour un TP classique. Des études théoriques et empiriques couvrant différents domaines d'enseignement et de larges populations d'apprenants, ont démontré que les télé-TPs en ligne peuvent être aussi efficaces que les TP si un certain nombre de principes sont respectés. Les études présentées ont conclu que les manipulations virtuelles sur des dispositifs distants ou simulés peuvent remplacer les manipulations physiques et permettre l'acquisition de compétences techniques requises pour le futur ingénieur ou chercheur. Pour augmenter les chances d'atteindre ces compétences, les manipulations virtuelles doivent prendre place dans un environnement virtuel fidèle à l'environnement du laboratoire. Non seulement en termes d'outils de commande techniquement réalistes, mais aussi en termes de reproduction du sentiment de présence et d'immersion dans le laboratoire distant ou virtuel.

Aussi, pour limiter le sentiment d'isolement des utilisateurs et faciliter l'acquisition de compétences sociales et communicatives, nous avons identifié des techniques issues de recherches en IHM et en TCAO/ACAO qui aident à construire un tel environnement ; en particulier, l'*Awareness* de l'espace de travail et l'*Awareness* de l'artefact qui visent à mettre tous les acteurs au même niveau d'information sur l'évolution des expériences.

Toujours dans la perspective d'atteindre la même efficacité pédagogique des travaux pratiques, mais cette fois par rapport aux activités d'apprentissage en ligne, nous nous sommes intéressés à l'intégration des télé-TPs dans les EIAH existants. L'intérêt de cette intégration est de tirer profit des avancées du domaine des EIAHs dans le domaine des activités pédagogiques en ligne. L'intégration des télé-TPs dans les EIAH existants suppose que ces deux ressources interagissent et offrent des interfaces normalisées qui assurent une communication transparente. Ceci nous renvoie aux normes et standards pédagogiques qui visent à rendre tout objet pédagogique repérable (via des normes de description telles que le LOM) et interopérable avec n'importe quel EIAH (via des normes d'agrégation comme IMS-CP et SCORM). D'autre part, pour assurer cette interaction, un protocole de communication standard dédié à la commande et à la supervision du laboratoire distant, ainsi qu'une modélisation ouverte et normalisée de ce laboratoire doivent être mis en œuvre.

Chapitre 3. Etat de l’art des télé-TPs

<u>3.1.</u>	<u>Le modèle d’analyse</u>	66
<u>3.2.</u>	<u>Le projet PEARL</u>	67
<u>3.3.</u>	<u>Le projet Lab@Future</u>	71
<u>3.4.</u>	<u>Le projet FORMID</u>	75
<u>3.5.</u>	<u>Le projet ICTT@LAB</u>	78
<u>3.6.</u>	<u>Le projet eINST</u>	81
<u>3.7.</u>	<u>Le projet RoboTeach</u>	85
<u>3.8.</u>	<u>Le projet VITELS</u>	89
<u>3.9.</u>	<u>Le projet Tele-Lab IT-Security</u>	92
<u>2.5.</u>	<u>Synthèse</u>	62

Notre problématique générale peut être reformulée à la lumière des éclairages théoriques apportés dans les deux précédents chapitres, comme étant l’intégration des télé-TPs dans les EIAHs existants pour assurer des formations complètement ou partiellement en ligne à des apprenants dans les disciplines de la science, de la technologie, de l’ingénierie et des mathématiques. Cette intégration doit préserver les avantages liés au déploiement de télé-TPs (la mutualisation et le partage des ressources entre institutions, la réduction des coûts d’acquisition et de maintenance des espaces d’hébergement, l’assouplissement et l’extension des horaires de télé-TPs), et assurer une efficacité pédagogique au moins équivalente à celle des TP traditionnels. Dans ce chapitre, nous étudions un certain nombre de projets traitant partiellement ou totalement la problématique énoncée ci-dessus, selon un modèle d’analyse issu des dimensions déterminantes pour l’efficacité pédagogique des télé-TPs qui ont été identifiées dans le chapitre précédent. Les résultats de cette analyse nous permettront d’identifier les verrous à lever pour construire un EIAH dédié aux télé-TPs.

3.1. Le modèle d'analyse

Nous avons construit un modèle d'analyse sur la base du modèle d'évaluation de l'efficacité pédagogique des télé-TPs proposé par (Ma et Nickerson, 2006), de la grille d'analyse de H. Benmohamed (Benmohamed, 2006), et du modèle de suivi synchrone de C. Després (Després, 2001) enrichi par les travaux de (Guéraud et al., 2004). Notre modèle d'analyse est divisé en sept catégories de critères pour lesquelles le Tableau 3-1 détaille les caractéristiques à identifier dans chacun des projets.

Tableau 3-1. Catégories des critères d'analyse des projets de télé-TPs.

Catégorie	Caractéristiques à identifier
Caractéristiques générales	Le type d'institution de formation visé, les objectifs globaux et spécifiques que le projet vise à atteindre, la population d'apprenants ciblée, et le(s) type(s) d'activités mis en œuvre.
Caractéristiques du laboratoire distant	Le type de technologies utilisé, et la manière avec laquelle les objectifs d'apprentissage fixés sont atteints. L'architecture mise en œuvre, et les moyens de télé-opération disponibles.
Caractéristiques des expériences	Le(s) type(s) des expériences, et les objectifs visés.
Caractéristiques de l'EIAH	Les moyens d'intégration des télé-TPs dans les EIAHs.
Conception pédagogique	Les outils de conception pour les télé-TPs, et le niveau auquel ils s'appliquent.
Support à l'apprentissage collaboratif	Les outils supports à l'apprentissage collaboratif en ligne, et la manière dont ils considèrent la collaboration.
Support aux tâches de tutorat	Les outils supports au tutorat en ligne, et la manière dont ils considèrent le tutorat.

Notre étude de l'existant se restreint aux projets de télé-TPs conçus avec une intention pédagogique, et ne prend pas en considération les projets de mise en ligne de laboratoires. Nous pouvons déjà les diviser en deux grandes catégories : les EIAHs génériques indépendants du domaine d'enseignement (ou les projets orientés pédagogie), et les EIAHs spécifiques à un domaine particulier (ou projets orientés performance). En ce qui concerne les projets génériques, nous en avons identifié cinq :

- Le projet PEARL (*Practical Experimentation by Accessible Remote Learning*) de l'Open University¹⁰ (Grande Bretagne).
- Le projet européen Lab@Future (*School LABORatory anticipating FUTURE needs of European Youth*).
- Le projet FORMID¹¹ (FORMation Interactive à Distance) de l'Université Joseph Fourier de Grenoble.
- Le projet ICTT@LAB (*generIC framework for remoTe and virTu@l Laboratory integration*) de l'INSA de Lyon.
- eINST¹² (*E-INSTRumentation Project*) de l'Ecole d'Ingénieurs Télécoms de l'Université Jean Monnet.

Les projets spécifiques identifiés sont les suivants :

- Le projet RoboTeach¹³ dédié à la robotique pédagogique, du laboratoire du LIUM¹⁴ à l'Université du Maine en France.
- Le projet VITELS¹⁵ (*The Virtual Internet and TELEcommunications Laboratory of Switzerland*) dédié aux réseaux et télécoms en Suisse.
- Le projet Tele-Lab IT-Security¹⁶ dédié à la sécurité informatique et issu du Hasso-Plattner-Institute (Allemagne).

3.2. Le projet PEARL

3.2.1. Caractéristiques générales

PEARL est un projet européen démarré en mars 2000 et dirigé par l'Open University en Grande Bretagne, en collaboration avec trois universités Européennes. Il vise le développement d'un système d'expérimentations réelles à distance dans un contexte de e-

¹⁰ <http://www.open.ac.uk/>

¹¹ <http://www-clips.imag.fr/arcade/projets/FORMID/>

¹² <http://diom.telecom-st-etienne.com/satin/einst/index.html>

¹³ http://gev.industrie.gouv.fr/article.php3?id_article=334

¹⁴ <http://www-lium.univ-lemans.fr/fr/content/bienvenue>

¹⁵ <http://www.vitels.ch/course/overview.php>

¹⁶ http://www.hpi.uni-potsdam.de/meinel/projekte/security_labs/tele_lab_it_security.html

formation où les apprenants doivent être capables d'interagir avec l'expérience à distance, de modifier les expériences, de discuter de leurs actions et prévisions, ainsi que d'observer et d'analyser les résultats à travers des outils de communication et de collaboration. L'expérimentation a été réalisée sur quatre domaines d'enseignement : la biochimie, la physique fondamentale, l'inspection visuelle de circuits imprimés PCB (*Printed Circuit Board*) et l'électronique (Cooper et al., 2000).

La Figure 3-1 illustre l'architecture globale du système PEARL que nous détaillons dans la suite. Elle est constituée de quatre composants (Pearl, 2000D4.1) :

- L'environnement d'enseignement virtuel correspond à la plateforme de télé-enseignement WebCT, et supporte les fonctions pédagogiques de conception, de tutorat et d'apprentissage.
- Les outils de collaboration synchrone et asynchrone pour supporter l'apprentissage collaboratif et la gestion de l'accès aux équipements du laboratoire.
- Le système d'administration permet de contrôler et de superviser le laboratoire distant et les utilisateurs qui l'exploitent.
- Le gestionnaire des équipements d'expérimentation responsable de leur configuration, contrôle et supervision, mais également du suivi des actions opérées par les utilisateurs.

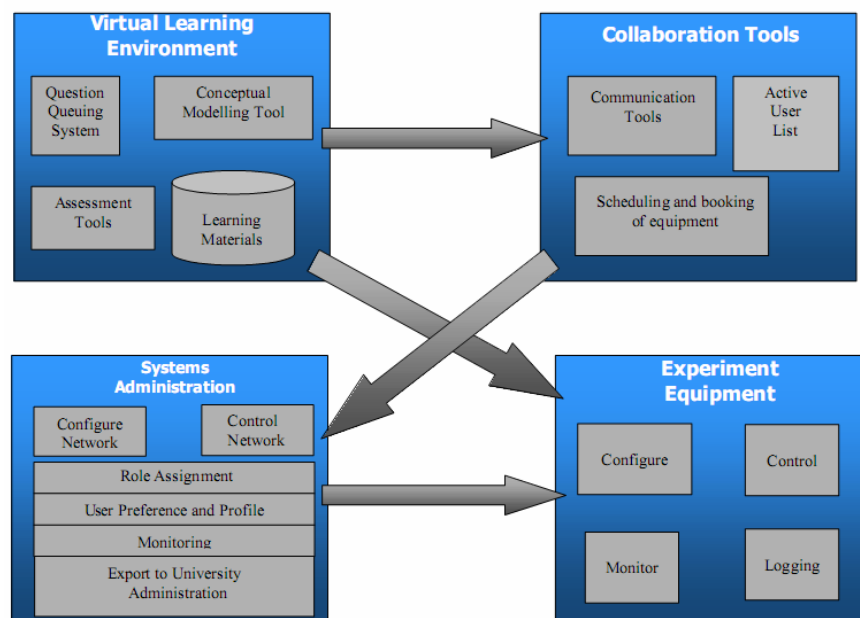


Figure 3-1. Architecture générale du système PEARL (Pearl, 2000D4.1).

3.2.2. Caractéristiques du laboratoire distant

Les laboratoires distants dans le projet PEARL sont constitués pour la plupart de dispositifs réels. Pour permettre la télé-opération des laboratoires distants, un serveur dédié à la médiatisation de l'instrumentation des équipements du laboratoire exploite tout type de technologies de bus en fonction de l'interface offerte par le dispositif contrôlé et la nature de l'expérience (comme RS232, USB, Ethernet, etc.). La technologie CORBA de l'OMG a donc été utilisée comme Middleware pour assurer la communication entre l'interface utilisateur et le laboratoire.

L'IHM d'instrumentation est développée sous la forme d'une Applet Java intégrée dans une page web, et accessible via la plateforme de télé-enseignement WebCT. Quatre interfaces ont été développées pour interagir avec quatre dispositifs de laboratoires associés aux quatre domaines d'enseignement testés, et dont l'IHM est adaptable au profil de l'apprenant.

La gestion des rôles et des profils ainsi que la politique d'accès sont prises en charge au niveau du serveur dédié aux tâches d'administration. La gestion de la réservation et de l'ordonnancement de l'accès aux équipements du laboratoire est assurée par un composant dédié au niveau du serveur de collaboration.

Enfin, chaque laboratoire a été conçu pour une expérience unique, mais il assure la séparation des deux concepts grâce à son méta-modèle de description générique qui permet de représenter tout type de laboratoire et d'expérience.

3.2.3. Caractéristiques de l'EIAH

Le système PEARL peut être considéré comme un EIAH générique intégré au LMS WebCT. L'interaction avec le laboratoire s'effectue à travers le moniteur du modèle du laboratoire, permettant ainsi d'enregistrer toutes les opérations effectuées sur les dispositifs distants. De plus, PEARL utilise des caméras pour superviser les dispositifs distants en plus des indicateurs affichés sur l'IHM de télé-opération.

L'IHM de télé-apprentissage dans PEARL intègre les fenêtres de télé-opération et de supervision du laboratoire distant, complétées par des ressources d'apprentissage et des outils de communication et de collaboration. Cette conception a été adoptée après une étude démontrant que l'ouverture de fenêtres indépendantes pour chaque tâche augmente la charge

cognitive des apprenants (Ferreira et al., 2002). La Figure 3-2 présente l'IHM d'expérimentation d'un télé-TP sur la programmation logique en électronique.



Figure 3-2. IHM d'expérimentation sur un système électronique (Ferreira et al., 2002).

3.2.4. Support à la conception pédagogique

PEARL est fondé sur la théorie socioconstructiviste, et se focalise sur les principes d'apprentissage actif et d'apprentissage par essais et erreurs (Cooper et Ferreira, 2009). Une activité de télé-TP est modélisée à travers un *LabScript* équivalent à un scénario IMS-LD, et guide l'apprenant jusqu'à ce qu'il atteigne le but de l'activité. Une activité est constituée d'une ou plusieurs sections représentant chacune une expérience sur le laboratoire (pour les détails, consulter (Pearl, 2001D8.1)). Grâce à ce modèle, les traces d'un télé-TP sont enregistrées dans un fichier HTML facilement consultable.

Les initiateurs du projet proposent des tutoriels, des modèles prêts à l'emploi et des lignes directrices pour faciliter à l'enseignant le développement de scénarios de télé-TPs. Le choix d'un modèle pédagogique approprié à partir d'un éventail de possibilités devrait encourager la réflexion sur la finalité des travaux prévus et sur la comparaison des différents styles envisageables de télé-TPs (Colwell et al., 2002).

3.2.5. Support à l'apprentissage collaboratif et au tutorat en ligne

L'apprentissage collaboratif est supporté à travers les outils de communication et de collaboration fournis par la plateforme WebCT (messagerie électronique, forum, dépôt de documents, etc.), et les outils fournis par le serveur de collaboration CuSeeME. Ce dernier offre des outils comme la visioconférence, la messagerie instantanée, la consultation

synchrone des pages web et le partage d'applications via le protocole T.120. En outre, une forme d'*Awareness* est implémentée à travers l'affichage dans l'IHM de la liste des apprenants actifs.

Le tuteur accède à tous les outils et espaces de communication, de collaboration et d'expérimentation accessibles à tous ses apprenants et groupes d'apprenants. Il dispose en plus d'outils qui lui sont spécifiques dédiés aux tâches suivantes : perception des activités des apprenants, suivi individuel et collectif des apprenants, et gestion du suivi des apprenants.

3.2.6. Synthèse du projet PEARL

Le projet PEARL est l'un des projets les plus aboutis que nous avons étudiés, car il couvre tous les éléments de notre modèle d'analyse. En effet, il prend en compte aussi bien les aspects techniques et technologiques relatifs au réalisme des expériences en ligne, que les théories d'apprentissage et les aspects pédagogiques en supportant toutes les fonctions de conception, de tutorat et d'apprentissage.

L'intégration entre télé-TPs et EIAHs existants est clairement établie à travers l'utilisation du langage universel et ouvert XML pour la modélisation des expériences, des activités d'apprentissage et des profils des acteurs humains. Nous pouvons déduire que la modélisation PEARL établit une distinction entre le dispositif du laboratoire, l'expérience qui peut faire appel à différents dispositifs, et l'activité qui structure le télé-TP. Un des points forts de PEARL est le traçage structuré des activités des apprenants à deux niveaux : le niveau de télé-opération sur le laboratoire, et le niveau global d'apprentissage.

3.3. Le projet Lab@Future

3.3.1. Caractéristiques générales

Le projet LAB@FUTURE (*School **LAB**oratory anticipating **FUTURE** needs of European Youth*) est un projet de recherche et développement financé par la commission européenne dans le cadre du programme 7^{ème} IST. Son objectif est de supporter et développer les pratiques de l'enseignement de l'ingénierie en laboratoires dans les écoles, collèges, lycées, et universités. LAB@FUTURE utilise des objets réels et virtuels interfacés avec des systèmes mécatroniques, des interfaces de réalité augmentée, des technologies mobiles et des

environnements 3D multiutilisateurs. L'amélioration pédagogique des aspects fonctionnels et conceptuels de ce système est soutenue par une approche d'opérationnalisation dialogique¹⁷ de plusieurs théories d'apprentissage que sont la théorie de l'apprentissage expansif issue de la théorie de l'activité, et le constructivisme (Baudin et al., 2003).

3.3.2. Caractéristiques du laboratoire distant

L'architecture générale du projet illustrée par la Figure 3-3 correspond à l'architecture générique présentée dans la section 2.4.3.1 dédiée au contrôle de dispositifs réels. Le serveur principal LAB@FUTURE gère les profils des utilisateurs et se coordonne avec les autres composants pour l'accès au télé-TP. Le serveur de laboratoire ESS (*Experiment Specific Server*) médiatise les dispositifs physiques ou virtuels utilisés dans l'activité, alors que le serveur GCCS (*Generic Communication and Collaboration Server*) fournit les outils et interfaces nécessaires à la communication et à la collaboration entre les acteurs de la formation. La spécificité de ce projet est l'introduction d'un serveur multiutilisateurs de réalité virtuelle qui assure la connexion entre les dispositifs physiques et les objets informatiques afin de maintenir la synchronisation dans le cadre d'une expérience en réalité mixte (Baudin et al., 2005).

¹⁷ Fusion ou utilisation conjointe de deux théories différentes

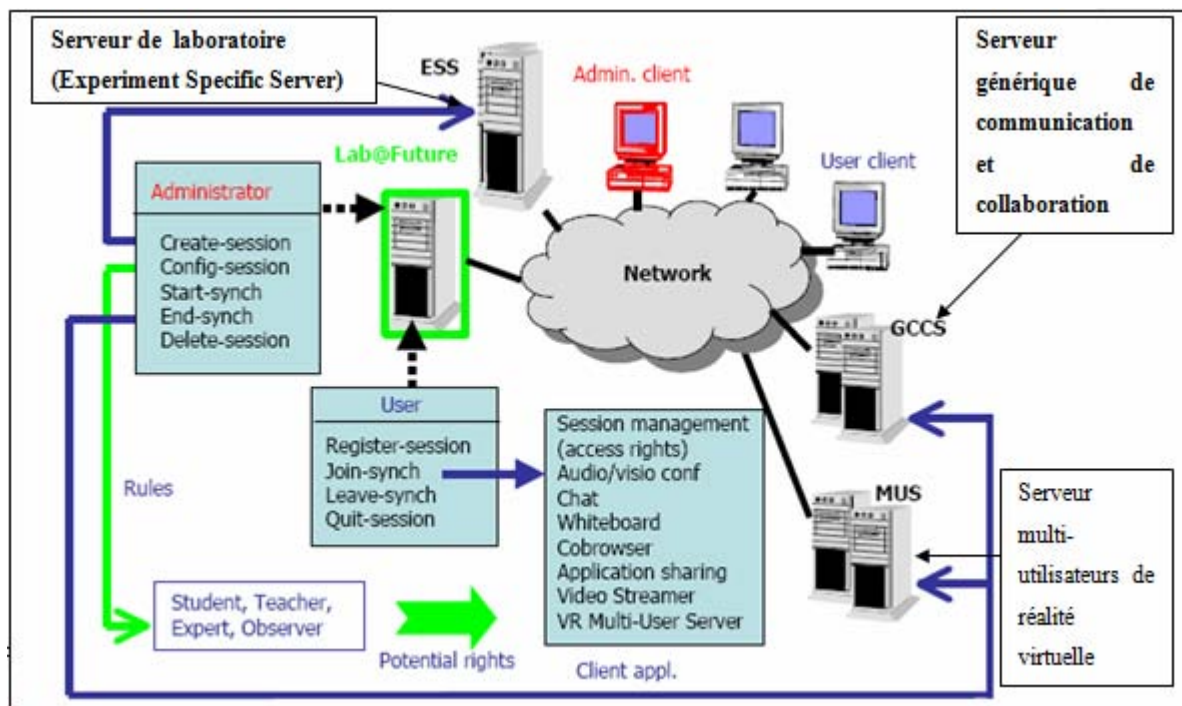


Figure 3-3. Architecture générale du projet LAB@FUTURE.

3.3.3. Caractéristiques de l'EIAH

Le système LAB@FUTURE forme un EIAH spécifique qui intègre tous les outils de l'enseignement et de l'apprentissage en ligne offerts par les plateformes de télé-enseignement. Mais il possède tous les ingrédients pour être interopérable avec d'autres EIAHs grâce à l'utilisation du langage XML pour la description des laboratoires, et du protocole HTTP pour la communication entre ses différents composants.

3.3.4. Support à la conception pédagogique

Le projet LAB@FUTURE est fondé sur les éléments de la théorie socioconstructiviste et ceux de la théorie de l'apprentissage expansif (Engeström, 1987). Cette dernière repose sur la théorie d'activité qui considère que les apprenants sont engagés consciemment ou inconsciemment dans la constitution d'objectifs dynamiques.

La scénarisation des télé-TPs dans LAB@FUTURE adopte la méthode AODM (*Activity Oriented Design Method*) élaborée par (Mwanza, 2002) qui permet de visualiser, sur le triangle d'Engeström (Engeström, 1987), tous les éléments constituant le télé-TP. Puis, en

suivant une démarche méthodique de questionnements, de réponses et d'analyses des contradictions, les auteurs conçoivent le scénario.

Bien qu'elle soit pertinente du fait qu'elle opérationnalise les éléments d'une théorie d'apprentissage, cette méthode est lourde pour la plupart des concepteurs et nécessite souvent la collaboration d'un expert de la théorie d'activité. Nous pensons qu'il vaut mieux formaliser cette méthode et créer le maximum de scénarios types réutilisables.

3.3.5. Support à l'apprentissage collaboratif en ligne

L'apprentissage collaboratif en ligne dans LAB@FUTURE reproduit l'apprentissage collaboratif en classe. D'une manière plus précise, LAB@FUTURE supporte l'apprentissage collaboratif via des outils de communication et de collaboration, ainsi que par des IHMs pour le support de l'*Awareness* de l'espace de travail et l'*Awareness* de l'artefact.

3.3.6. Support au tutorat en ligne

Dans LAB@FUTURE, le tuteur est un acteur immergé dans les environnements de réalités mixte et virtuelle offerts aux apprenants. Il pourra donc effectuer les mêmes tâches de suivi et d'évaluation qu'il effectue dans un laboratoire réel en utilisant les outils et IHMs décrits plus haut.

3.3.7. Synthèse

LAB@FUTURE se veut un projet innovant de mise en ligne des travaux pratiques, aussi bien du côté pédagogique que technologique. Par rapport à l'aspect pédagogique, une étude profonde a été engagée, fondée sur les éléments de deux théories d'apprentissage : le socioconstructivisme et l'apprentissage expansif. La méthode AODM qui opérationnalise la théorie de l'activité a été utilisée pour la conception d'un environnement et d'un scénario de télé-TP spécifique à un domaine donné. D'un point de vue pratique, cette méthode est assez lourde et nécessite l'implication d'un expert. Il serait plus judicieux de former des groupes de travail qui construisent des scénarios modèles (ou « *Best Practices* ») pour chaque domaine d'enseignement, et de les mettre à disposition des concepteurs afin de les réutiliser ou de les adapter au contexte de chaque situation.

Par rapport à l'aspect technologique, LAB@FUTURE introduit des technologies de pointe pour la construction d'IHMs de réalités mixte et virtuelle. Ceci favorise l'atteinte de l'efficacité pédagogique des télé-TPs en proposant des conditions de travail similaires à celles d'un TP classique, et en réifiant des concepts théoriques. Toutefois, l'intégration avec les EIAHs existants n'est pas possible par manque de normalisation des objets pédagogiques issus de la réalité virtuelle et de la réalité mixte.

3.4. Le projet FORMID

3.4.1. Caractéristiques générales

Le projet FORMID (FORMation Interactive à Distance) représente l'évolution de plusieurs travaux successifs : l'outil auteur OASIS¹⁸ fondé sur le modèle MARS¹⁹ (Pernin, 1996) pour la conception et la réalisation de simulations pédagogiques, et l'architecture ARGOS²⁰ (Buitrago, 1999) pour lier une simulation à un contrôle pédagogique. Les travaux de V. Guéraud sur la modélisation et l'exploitation pédagogique de situations actives d'apprentissage utilisant des objets pédagogiques interactifs (OPI) de type simulation sont au cœur de ce projet (Guéraud et al., 2003 ; Guéraud et Cagnat, 2004 ; Guéraud, 2005 ; Guéraud et al., 2004).

FORMID « vise à mettre en œuvre des situations actives d'apprentissage (telles que les travaux dirigés ou les travaux pratiques) dans les systèmes de FOAD. Il s'agit de donner aux formateurs les moyens de contrôler l'activité des apprenants, et de donner aux différents participants les moyens de communiquer entre eux »²¹. Les domaines enseignés varient et visent les publics des écoles, collèges et lycées.

3.4.2. Caractéristiques du laboratoire distant

Le projet FORMID est centré sur les simulations pédagogiques donc les laboratoires virtuels. Les simulations pédagogiques, stockées dans une base de données, sont téléchargées à partir

¹⁸ Outil Auteur de Simulations Interactives avec Scénarios

¹⁹ Modèle Association Représentation Scénario

²⁰ Architecture Générale pour l'Observation et le contrôle pédagogique de Simulation

²¹ <http://www-clips.imag.fr/arcade/projets/FORMID/>

du serveur FORMID vers le navigateur de l'utilisateur pour les exécuter au sein d'une applet Java. Le serveur FORMID correspond au serveur de laboratoire dans l'architecture générique du chapitre 2, où chaque apprenant exploite une instance de la même simulation dans son navigateur.

Les simulations sont définies selon un format textuel qui décrit la liste des variables d'état qui permettent de manipuler le programme de la simulation par un contrôle pédagogique. Même si l'IHM d'instrumentation est spécifique à chaque simulation, l'apprenant dispose d'une interface générique de type classe virtuelle. Cette interface intègre, en plus de la simulation, les autres ressources d'apprentissage et les outils de communication et de collaboration. Les simulations étant exécutées au sein du navigateur de l'apprenant, elles ne nécessitent pas de procédures de réservation particulières.

3.4.3. Caractéristiques de l'EIAH

FORMID est un EIAH spécifique à part entière qui implémente deux types d'interfaces : le poste de l'apprenant pour toutes les activités d'apprentissage, et le poste d'enseignant pour toutes les activités de tutorat. Toutefois, le fait d'utiliser des modèles ouverts pour la modélisation des simulations et la conception pédagogique rend possible l'intégration des laboratoires FORMID dans n'importe quel EIAH.

3.4.4. Support à la conception pédagogique

Un télé-TP dans FORMID est modélisé sous la forme d'un scénario IMS-LD qui caractérise l'activité en termes de titre, d'énoncé général et d'objectifs globaux. Une activité est constituée d'une suite d'étapes significatives correspondant chacune à des objectifs spécifiques à atteindre (Guéraud et al., 2004), le passage d'une étape à une autre étant conditionné par l'atteinte des objectifs de l'étape en cours.

Le projet est donc fondé sur les éléments de la théorie d'apprentissage constructiviste et la théorie de l'activité, et introduit la notion générale de situation active d'apprentissage « dans lesquelles l'apprenant interagit étroitement avec un dispositif l'induisant à une découverte-construction de connaissances » (Guéraud et al., 2004). La conception et la réalisation des simulations sont supportées par l'outil OASIS, qui a été étendu pour la conception de scénarios pédagogiques exploitant les objets pédagogiques interactifs (Guéraud, 2005).

3.4.5. Support à l'apprentissage collaboratif et au tutorat en ligne

La plateforme semble être dédiée à un apprentissage autonome guidé par le scénario pédagogique et soutenu par un tuteur distant selon les principes de la théorie constructiviste qui exclut l'apprentissage collaboratif. Afin d'assurer un suivi efficace aux apprenants distants, le moniteur de scénarios enregistre, dans une base de données dédiée, les états des simulations exploitées. Ainsi le tuteur dispose en temps réel de l'état d'avancement de chaque apprenant pour chaque activité et pour chaque sous-étape. La Figure 3-4 illustre l'interface de la classe virtuelle de la plateforme FORMID.



Le développement de ces interfaces est fondé sur le modèle de suivi individuel synchrone des apprenants SAAD de C. Desprès (Desprès, 2001), et étendu au suivi collectif d'une classe virtuelle (Guéraud et al., 2004). Le tuteur peut ainsi apporter un soutien aux apprenants via un outil de messagerie instantanée, et exécuter dans son interface la simulation. Toutefois, il ne peut pas partager la simulation de l'apprenant (absence de l'*Awareness* de l'artefact).

Enfin, FORMID permet au tuteur de gérer les messages de sollicitation des apprenants. C'est un mécanisme qui ressemble à la queue de messages dans PEARL, et qui aide à diminuer la charge cognitive sur le formateur en lui permettant de filtrer les messages des apprenants par ordre chronologique, par priorité, etc.

3.4.6. Synthèse

Le projet FORMID constitue la synthèse d'une longue recherche en EIAH pour la mise en ligne de travaux pratiques fondés sur les simulations. FORMID a pu évoluer grâce à la souplesse et la simplicité des simulations pédagogiques qui constituent le cœur des activités qu'il propose. En effet, en éliminant les problèmes de télé-opération posés dans le contexte de laboratoires réels, il a pu se focaliser sur des aspects pédagogiques pertinents tels que la scénarisation de situations actives d'apprentissage, les aspects de l'IHM et le suivi en ligne des classes virtuelles.

Parmi les résultats importants de ce projet, notons la définition d'un ensemble de services standards pour l'interopérabilité des objets pédagogiques interactifs avec les plateformes d'apprentissage, et en particulier avec les scénarios d'apprentissage IMS-LD. Grâce à ces standards ouverts, FORMID assure également un suivi structuré des différentes opérations sur les simulations.

L'inconvénient majeur du projet est l'absence de soutien à l'apprentissage collaboratif, pourtant identifié comme élément essentiel à l'efficacité des télé-TPs. Par conséquent, il n'est pas possible de favoriser l'acquisition de compétences sociales.

3.5. Le projet ICTT@LAB

3.5.1. Caractéristiques générales

Le projet ICTT@LAB (*generIC framework for remoTe and virTu@l LABoratory integration*) est le résultat des travaux de thèse de H. Benmohamed (Benmohamed, 2007) qui repose sur deux projets antérieurs (PIEUVRE et TIPY²²) développés au laboratoire LIESP (ex. ICTT) de l'INSA de Lyon. Le public cible concerne les étudiants de premier cycle des écoles d'ingénieurs et des universités.

ICTT@LAB est un cadre de travail informatique qui permet l'intégration des télé-TPs dans les environnements de télé-formation existants comme les LMSs, tout en offrant une chaîne d'édition de scénarios pédagogiques de type télé-TP. Ainsi toutes les fonctions classiques de

²² <http://elabs.insa-lyon.fr/>

gestion et de diffusion de ressources d'apprentissage, de gestion des enseignants et des apprenants, et de gestion et de planification de l'apprentissage sont assurées au niveau de l'EIAH d'origine grâce à une couche intermédiaire qui assure une intégration transparente de l'EIAH et de la plateforme de télé-TP (Benmohamed, 2006).

3.5.2. Caractéristiques du laboratoire distant

Le projet ICTT@LAB repose sur une architecture trois-tiers dont l'objectif est de faciliter l'intégration de laboratoires distants dans n'importe quel LMS. La

Figure 3-5 illustre tous les composants de l'EIAH ICTT@LAB répartis en trois niveaux :

- Le niveau télé-formation implémente les fonctions pédagogiques d'apprentissage, de tutorat et de conception.
- Le niveau télé-opération concerne la mise en ligne de laboratoires distants.
- Le niveau interface assure la liaison entre l'interface utilisateur et le laboratoire distant.

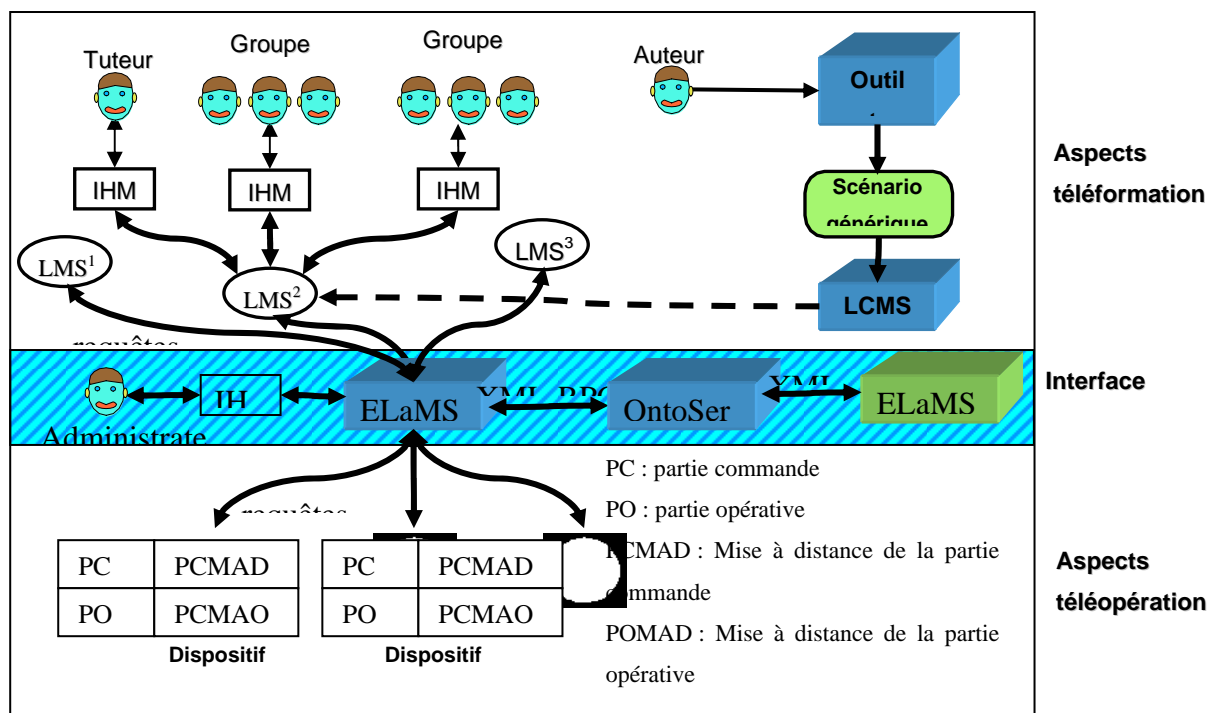


Figure 3-5. Architecture générale du projet ICTT@LAB (BenMohammed, 2006).

Cette architecture assure la prise en charge de tous les types de laboratoires, mais le projet se focalise sur la mise en ligne de laboratoires physiques et utilise les interfaces d'instrumentation disponibles. Le laboratoire distant est modélisé via une ontologie OWL composée de deux parties : la partie commande qui représente les moyens d'interaction avec le dispositif (commandes et retours), et la partie *opérative* qui décrit le fonctionnement interne du dispositif. Tout comme les projets PEARL et FORMID, l'objectif de cette modélisation est l'intégration dans les scénarios pédagogiques IMS-LD. La gestion et la réservation des ressources sont assurées par un outil intégré au serveur ELaMS.

3.5.3. Caractéristiques de l'EIAH

Comme dans le projet PEARL, le projet ICTT@LAB utilise un LMS *open source* existant : Moodle. L'avantage de Moodle est sa souplesse et son extensibilité qui permettent d'intégrer des composants ou *plugins* spécifiques à des tâches particulières. Le LMS ne communique pas directement avec les dispositifs distants, mais à travers une interface au sein du serveur qui opérationnalise le modèle du laboratoire. Le télé-TP est modélisé et implémenté sous la forme d'un scénario IMS-LD exécuté par l'outil CopperCore²³.

3.5.4. Support à la conception pédagogique

ICTT@Lab préconise l'outil auteur « Protégé » pour la modélisation des dispositifs de laboratoires et la production des modèles OWL correspondants, et l'outil auteur Reload²⁴ pour la scénarisation des télé-TPs.

3.5.5. Support à l'apprentissage collaboratif et au tutorat en ligne

Le support à l'apprentissage collaboratif et au tutorat en ligne est assuré par le niveau télé-formation, c'est-à-dire le serveur Moodle. Notons que le projet ne couvre pas les aspects de l'*Awareness* de l'espace de travail ni ceux de l'*Awareness* de l'artefact.

²³ <http://coppercore.sourceforge.net/>

²⁴ <http://www.reload.ac.uk/ldeditor.html>

3.5.6. Synthèse

Les travaux de H. Benmohamed sur les télé-TPs visent à résoudre la problématique d'intégration de ces activités dans les plateformes d'apprentissage. Le problème est abordé sous l'angle de la normalisation de la e-formation, et plus particulièrement la normalisation des activités d'apprentissage. Ainsi, un laboratoire est modélisé en utilisant les modèles du web sémantique, et les modèles générés peuvent être manipulés par des scénarios IMS-LD. L'auteur propose une chaîne d'édition complète destinée à l'enseignant concepteur allant de la modélisation des dispositifs de laboratoire à la modélisation des scénarios de télé-TPs, en passant par la sauvegarde et la publication de ces modèles au profit de la communauté. Mais le projet ne traite pas encore les problématiques de l'apprentissage collaboratif et du tutorat en ligne.

3.6. Le projet eINST

3.6.1. Caractéristiques générales

eINST²⁵ (**E-INST**strumentation Project) est un projet de laboratoire collaboratif distant issu des travaux de thèse de C. Gravier dont l'objectif est de permettre l'apprentissage collaboratif dans les sessions de télé-TPs, et de rendre réutilisables et génériques les dispositifs technologiques distants servant ces télé-TPs (Gravier, 2007). eINST s'adresse aux étudiants universitaires de premier cycle pour les cursus en sciences et ingénierie.

Le projet eINST est fondé sur les mêmes principes que le projet ICTT@Lab en ce qui concerne l'utilisation des ontologies web pour la modélisation formelle des laboratoires, mais il traite en plus les problématiques de conception de l'IHM de télé-opération (et son interconnexion avec les dispositifs distants), de gestion des traces des apprenants, et de support de l'apprentissage collaboratif en ligne.

²⁵ <http://diom.telecom-st-etienne.com/satin/einst/index.html>

3.6.2. Caractéristiques du laboratoire distant

L'architecture générale du projet eINST illustrée par la Figure 3-6 est caractérisée par l'utilisation d'un Middleware Orienté Messages entre l'interface utilisateur distant et l'appareil télé-opéré.

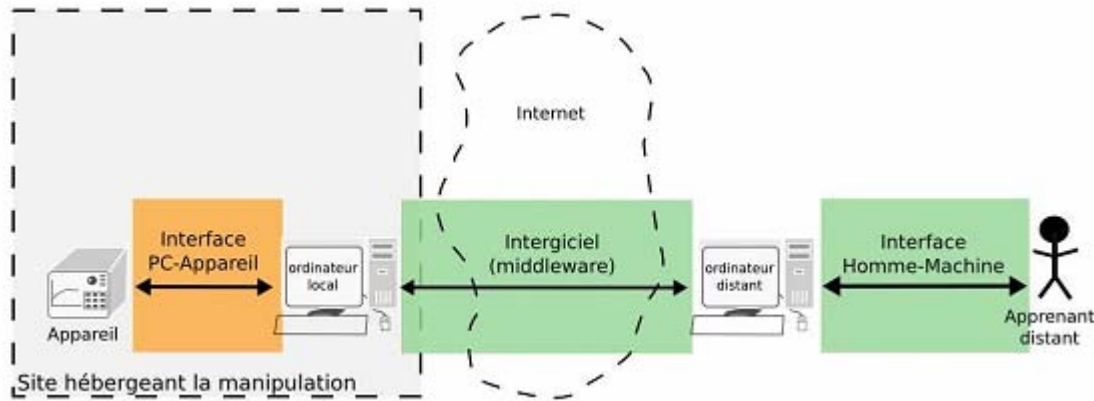


Figure 3-6. Architecture générale du projet eINST.

Les laboratoires sont modélisés en utilisant le langage OWL dédié aux ontologies web, mais ces modèles se limitent à la formalisation de l'IHM du dispositif distant qui correspond dans le projet ICTT@Lab à la « Partie Commande » (voir la

Figure 3-5). Cette IHM est construite à partir du modèle OWL du dispositif où chaque élément représente un *Widget* (par exemple un bouton). Outre sa simplicité de mise en œuvre, une conséquence positive de cette modélisation est la possibilité d'adapter l'IHM selon le niveau de l'apprenant (avancé, moyen, novice), ses préférences ou ses objectifs, en masquant certains *Widgets* de l'IHM. Par contre, des questions se posent quant à la faisabilité de la connexion du dispositif distant à une application de contrôle pédagogique comme les moniteurs IMS-LD, et par conséquent son intégration dans un scénario IMS-LD de télé-TP.

En ce qui concerne les politiques d'accès, eINST implémente un système sophistiqué pour orchestrer l'accès aux dispositifs distants. C'est un système fondé sur une base de données de règles d'accès exploitée par un moteur d'inférence qui détermine à un moment donné si l'acteur qui a demandé la main sur un dispositif distant peut l'exploiter ou non. L'idée du système est pertinente dans le sens où il va permettre une régulation intelligente de l'accès aux dispositifs distants parmi les acteurs humains. Son inconvénient est sa complexité qui va

nécessiter une longue expérimentation avant de retrouver des bases de règles de référence associées à différents types de politiques.

3.6.3. Caractéristiques de l'EIAH

Bien que le projet eINST propose un EIAH à part entière, les propriétés des modèles de laboratoires, des IHMs proposées et du Middleware de communication de tout le système représentent une barrière pour l'intégration de cet EIAH dans une plateforme de télé-enseignement, mais également pour la modélisation des activités de télé-TPs avec un langage standardisé tel qu'IMS-LD.

3.6.4. Support à la conception pédagogique

eINST propose un outil d'édition graphique de laboratoires distants spécifiques aux dispositifs de type façade (généralement des instruments de mesure, générateurs de fonctions, etc.). L'outil produit un fichier contenant (Gravier, 2007) : « Les médias (images et sons) utilisés par les *Widgets* (par exemple l'image d'un bouton ou encore le son joué lorsque l'on atteint le pas de rotation d'un bouton rond) » et « l'ontologie elle-même de la façade de l'instrument, au format OWL ». A partir d'une photo prise de l'équipement en question, l'outil applique des techniques de reconnaissances de contours pour identifier les composants classiques (boutons, leds, écrans d'affichage), et construit automatiquement l'ontologie correspondante. Le résultat n'étant jamais parfait, il est possible de reprendre l'ontologie et la corriger manuellement, mais ce processus simplifie et accélère les phases de modélisation et de génération de l'IHM du dispositif distant.

3.6.5. Support à l'apprentissage collaboratif en ligne

eINST prétend renforcer l'expérimentation collaborative en ligne via son système de régulation d'accès aux dispositifs distants présentés plus haut. Mais de notre point de vue, c'est un système qui va assurer un accès équilibré au dispositif parmi les membres d'un groupe engagés dans le même télé-TP, sans proposer de moyens spécifiques favorisant la collaboration.

eINST propose d'autre part un outil de messagerie instantanée intégré à l'IHM du télé-TP, et implémente une forme basique d'*Awareness* sous la forme d'un indicateur de présence des différents acteurs.

3.6.6. Support au tutorat en ligne

Le système eINST sauvegarde les traces d'interactions des apprenants avec les dispositifs distants dans une base de données, ce qui en facilite l'exploitation par la suite. Selon l'auteur, la journalisation vise deux objectifs :

- Elle permet de remettre l'appareil distant à son dernier état après la réexécution des actions passées qui ont été sauvegardées, sachant que l'appareil peut être éteint ou utilisé par d'autres apprenants dans le cadre d'un autre télé-TP.
- Elle permet de rejouer une session d'expérimentation a posteriori pour différentes raisons : démonstration, évaluation, etc.

3.6.7. Synthèse

eINST est un projet de télé-TP générique qui propose des solutions pertinentes à la problématique de mise en ligne de laboratoires distants. Bien qu'il propose une modélisation formelle standard et ouverte des dispositifs distants, l'approche d'eINST consiste à modéliser l'IHM de commande des appareils afin que le système construise automatiquement l'IHM d'exploitation du dispositif. En offrant un outil graphique d'édition des modèles et des IHMs associées, il simplifie grandement la tâche à l'enseignant concepteur. L'inconvénient de cette solution est qu'elle est limitée aux dispositifs de laboratoires commandés via une IHM de type façade.

En ce qui concerne l'apprentissage collaboratif, le projet s'est focalisé sur le problème de la régulation des accès concurrents au même dispositif distant par un groupe d'apprenants, mais il est regrettable que les autres aspects de l'apprentissage collaboratif et de tutorat en ligne n'aient pas été adressés.

3.7. Le projet RoboTeach

3.7.1. Caractéristiques générales

RoboTeach est un logiciel d'apprentissage des bases fondamentales en technologie et robotique. Développé par Pascal Leroux, du Laboratoire d'Informatique de l'Université du Mans (LIUM), il s'appuie sur les kits "Fischertechnik"²⁶ pour aider, pas à pas, les élèves à construire toutes sortes de maquettes d'équipements pilotés par ordinateur.

Au départ conçu comme un EIAO (monoposte) de type micro-monde utilisant le langage LOGO (Papert, 1981) pour le contrôle des robots pédagogiques, le projet a pris deux directions : l'EIAO RoboTeach a continué à évoluer pour être industrialisé et diffusé à grande échelle dans les écoles, et en parallèle une autre branche du logiciel a été développée pour satisfaire au besoin de la formation à distance produisant un Environnement Interactif d'Apprentissage à Distance (Bruillard et al., 2000 ; Leroux, 2002).

3.7.2. Caractéristiques du laboratoire distant

RoboTeach est fondé sur des dispositifs physiques pédagogiques, et présente deux types d'IHMs d'instrumentation : l'IHM de pilotage visuel et l'IHM de programmation et de pilotage de micro-robots. Le premier type est destiné à familiariser l'apprenant avec la manipulation des robots par une prise en main progressive de ces derniers (partie droite de la Figure 3-7), alors que la seconde IHM permet la programmation de ces robots via un langage de programmation simplifié (partie gauche de la Figure 3-7) (Leroux, 1997).

²⁶ <http://www.fischertechnik.de>

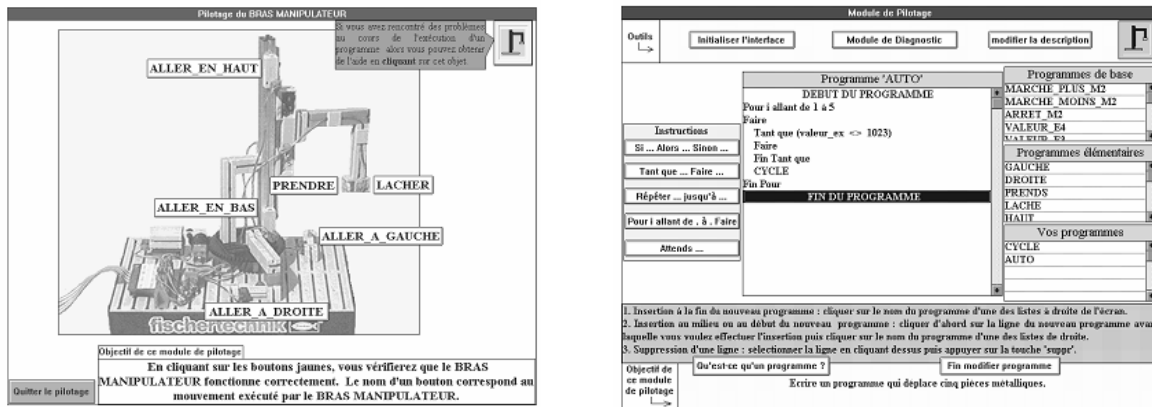


Figure 3-7. Deux IHMs de pilotage de RoboTeach (Leroux, 1997).

3.7.3. Caractéristiques de l'EIAH

RoboTeach est un EIAH spécifique qui intègre les résultats des différents travaux de recherche menés au LIUM durant plus de deux décennies sur l'assistance pédagogique apportée aux apprenants en classe (l'EIAO RoboTeach), les outils supports à l'apprentissage collaboratif (l'environnement SPLACH (George, 2001)), et les outils supports au tutorat (l'environnement ESSAIM (Desprès, 2001)). Malheureusement, nous n'avons pas pu obtenir d'informations sur la modélisation des dispositifs télécommandés, ni sur les activités pédagogiques proposées.

3.7.4. Support à la conception pédagogique

RoboTeach est fondé sur les théories d'apprentissage constructiviste et socioconstructiviste qui sont mises en œuvre à travers l'implémentation de différents types de stratégies pédagogiques : apprentissage par essais et erreurs, apprentissage par découverte active, résolution de problème, apprentissage par l'action et projets technologiques (Leroux, 2002).

Le projet propose une méthodologie de conception d'activités d'apprentissage appelée ATRIUM (Bruillard et al., 2000 ; Leroux, 2002), fondée sur une projection sur trois axes : les compétences visées, les activités associées à chaque compétence, et les outils nécessaires pour l'accomplissement de chaque activité.

Le projet s'est alors orienté vers des activités fondées sur la pédagogie par projet, et (George, 2001) a proposé dans l'environnement SPLACH un modèle général d'organisation d'une activité de projet distribuée qui divise l'activité en une succession d'étapes individuelles

asynchrones et collectives synchrones. L'activité démarre par un énoncé du projet à réaliser et se termine par le compte rendu de la réalisation. Le passage d'une phase à une autre au sein de l'activité est conditionné par la production d'un compte rendu intermédiaire.

3.7.5. Support à l'apprentissage collaboratif en ligne

Le support de l'apprentissage collaboratif en ligne est implémenté par l'environnement SPLACH. C'est un environnement qui intègre l'IHM de RoboTeach et des espaces et outils de communication et de collaboration. Les apprenants seront guidés par SPLACH d'étape en étape jusqu'à atteindre le but de l'activité. Ainsi, SPLACH intègre en une seule interface les outils suivants (George, 2001) :

- Un outil de documentation collaboratif pour la rédaction collaborative synchrone de comptes rendus.
- Un outil de planification qui se présente sous la forme d'un calendrier partagé entre les membres de l'équipe.
- Un outil de réunion synchrone fondé sur la messagerie instantanée, mais qui permet la régulation et la coordination des actions entre les membres du groupe.
- Des outils de communication asynchrone (messagerie électronique et forum).
- Des outils spécifiques au domaine d'apprentissage.

3.7.6. Support au tutorat en ligne

Le support au tutorat en ligne dans RoboTeach est implémenté par l'Environnement de Suivi pédagogique Synchrone d'Activités d'Apprentissage (ESSAIM), qui met en œuvre le modèle d'Activités d'Apprentissage à Distance (SAAD - Desprès, 2001). Ce modèle est divisé en trois composants (présentés dans la Figure 3-8) qui représentent les principales activités du tuteur en ligne. L'environnement ESSAIM implémente les outils qui supportent ces activités au niveau de l'interface apprenant pour lui permettre de solliciter le tuteur, et au niveau de l'interface du tuteur afin de superviser les actions de l'apprenant et de reconstituer sa progression dans l'activité d'apprentissage ; les détails de ces outils peuvent être consultés dans (Desprès, 2001). Nous pouvons remarquer que la collaboration est un sous-ensemble de ce modèle à travers les outils de communication et de collaboration offerts, et des différents espaces mettant en œuvre divers types d'*Awareness*.

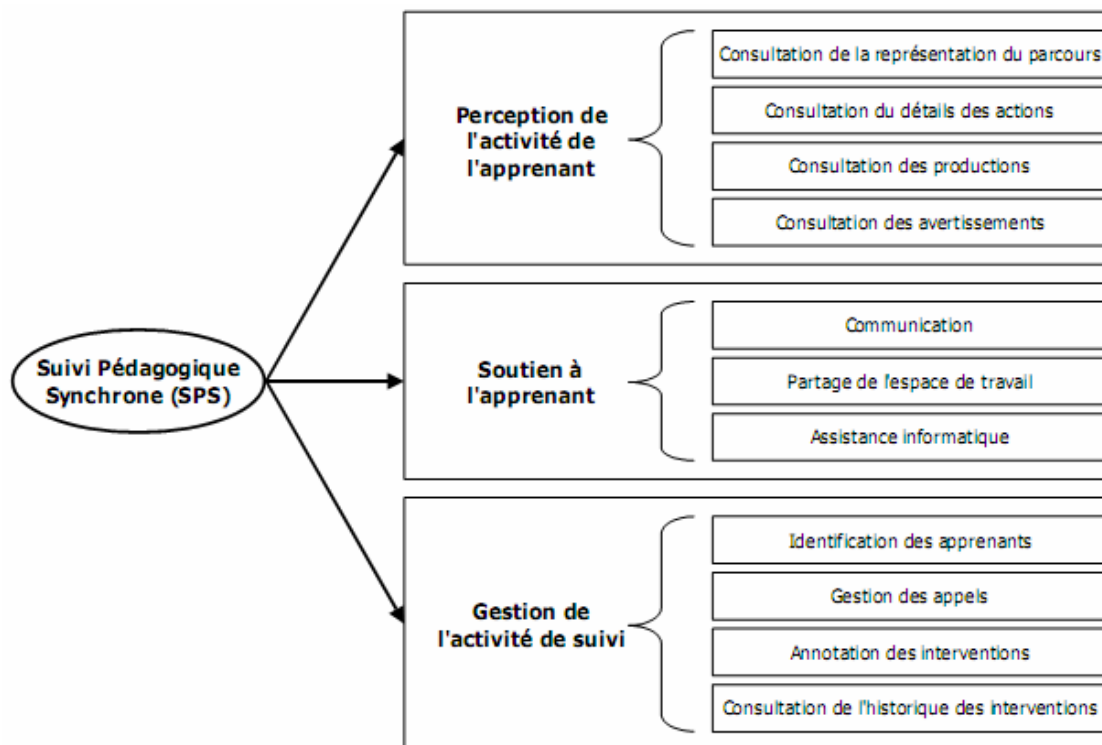


Figure 3-8. Le modèle de Suivi d'Activités d'Apprentissage à Distance (SAAD - Desprès, 2001).

3.7.7. Synthèse

Le projet RoboTeach a évolué avec le domaine des EIAHs. Nous avons distingué trois grandes étapes:

- Le développement de l'assistant pédagogique RoboTeach qui intègre l'interface de télé-opération des micros-robots et des kits de constructions, assortis des ressources pédagogiques adéquates. Il constitue le cœur de tout le système.
- L'environnement d'apprentissage collaboratif en ligne (SPLACH) : l'EIAH étant destiné à un domaine industriel, le travail collaboratif pour des groupes engagés dans des projets a été adopté comme stratégie pédagogique afin de préparer les apprenants aux conditions réelles de travail. L'environnement SPLACH implémente l'approche pédagogique d'apprentissage par projet, et offre aux apprenants un environnement qui intègre à la fois les fenêtres de télé-commande et de programmation des dispositifs physiques, mais aussi les outils et fenêtres de communication et de collaboration.

- L'environnement de tutorat en ligne : pour assister le tuteur en ligne, un environnement générique dédié a été développé (ESSAIM) qui implémente les éléments du modèle de suivi pédagogique des activités d'apprentissage à distance (SAAD). Grâce à cet environnement, le tuteur peut constituer une image précise de l'état d'avancement des apprenants à travers la perception synthétique et détaillée de leurs activités, apporter un soutien adéquat et en temps réel, et assurer une gestion efficace des sollicitations des apprenants en situation délicate.

Bien que le projet apporte des réponses pertinentes aux problématiques de mise en ligne de télé-TPs, il reste un environnement spécifique au domaine de la robotique pédagogique. La plupart de ses interfaces sont des applications clientes autonomes alors que la tendance actuelle s'oriente vers les navigateurs web pour la facilité et la sécurité de leur usage et maintenance. D'autre part, il n'existe pas d'intégration entre les environnements d'apprentissage collaboratif et celui du tutorat en ligne, alors qu'il est établi que les activités d'apprentissage collaboratif forment un sous-ensemble des activités de tutorat. D'ailleurs, les mêmes outils et fenêtres dédiés à la communication et à la collaboration, sont implémentés indépendamment dans chaque environnement.

Enfin l'environnement RoboTeach est un projet propriétaire dont les différentes implémentations ne sont pas exploitables ni intégrables dans d'autres EIAH. Aujourd'hui, différentes plateformes gratuites et ouvertes de développement d'applications de contrôle de robots existent indépendamment de leurs constructeurs, parmi lesquels URBI²⁷ et OROCOS²⁸.

3.8. Le projet VITELS

3.8.1. Caractéristiques générales

Le projet VITELS (*The Virtual Internet and Telecommunications Laboratory of Switzerland*) est un environnement de cours Internet orienté travaux pratiques. Le public cible est composé d'étudiants en informatique qui ont obtenu leurs deux premières années, ainsi que d'étudiants

²⁷ <http://www.gostai.com/>

²⁸ <http://www.oroocos.org/>

des universités techniques. VITELS tente de fournir des connaissances à travers les approches par essais et erreurs en utilisant les concepts pédagogiques constructivistes où l'étudiant apprend tout seul, n'importe où et n'importe quand (VITELS, 2004).

La gestion des modules d'enseignement est répartie sur l'ensemble des universités partenaires du projet. Chaque université gère et maintient un ou plusieurs modules, chacun intégrant une ou plusieurs expériences pratiques fondées sur des laboratoires réels, émuloés ou simulés. Cette division du travail permet de mutualiser les ressources des universités, et d'économiser des coûts d'acquisition de matériels et de logiciels, et surtout d'exploitation. Chaque module (et donc laboratoire) ajouté par un partenaire sera utilisé par les apprenants de toutes les universités du projet (VITELS, 2004).

3.8.2. Caractéristiques du laboratoire distant

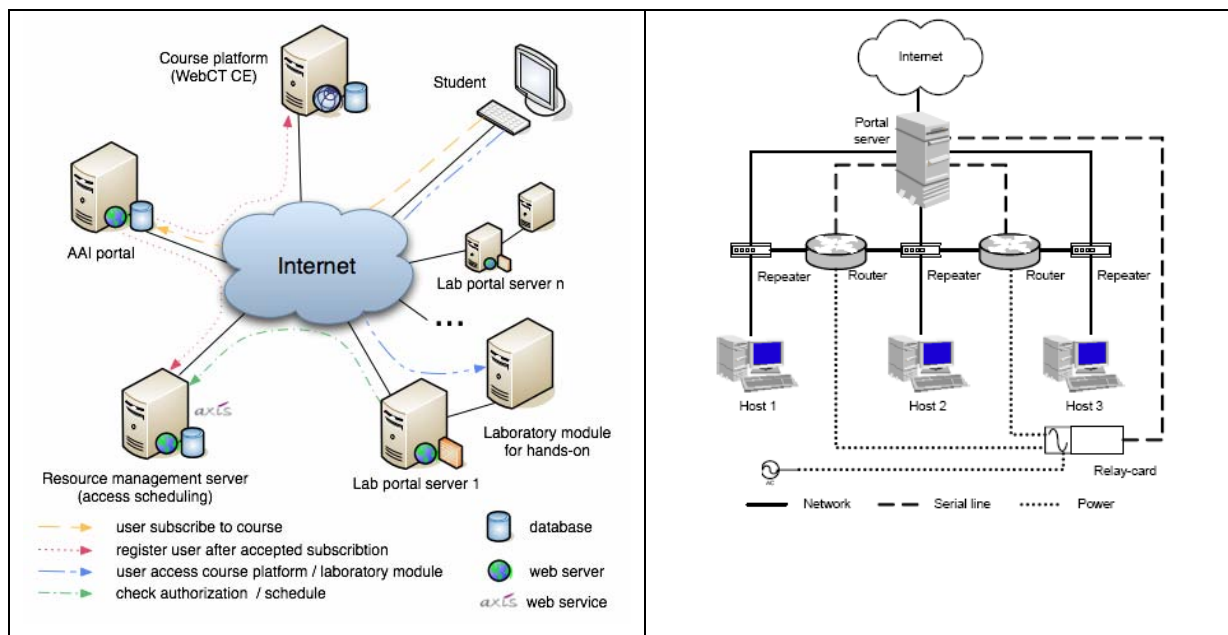


Figure 3-9. A gauche l'architecture générale du projet VITELS, à droite un exemple de laboratoire mis en ligne.

VITELS exploite des laboratoires réels ou émuloés, ainsi que des simulations, mais tous doivent se conformer à une architecture commune afin de pouvoir être partagés et exposés via la même plateforme de télé-enseignement WebCT. Cette architecture présentée par la Figure 3-9, se caractérise par l'utilisation d'un serveur appelé serveur portail pour médiatiser le laboratoire. La Figure 3-9 illustre également comment un laboratoire classique est transformé pour le rendre accessible en ligne. Afin de favoriser le principe de l'apprentissage par essais et

erreurs, VITELS implémente un mécanisme de réinitialisation à distance du laboratoire à travers un mécanisme électrique de mise hors/sous tension télé-commandé via le serveur portail (voir partie droite de la Figure 3-9).

Un laboratoire n'est pas modélisé, mais une représentation graphique est intégrée dans une page web afin d'en illustrer l'architecture réseau. D'autre part, il existe deux types d'interfaces pour interagir avec les laboratoires distants : l'interface ligne de commande via le protocole SSH, et l'interface graphique via le protocole VNC ; ces interfaces nécessitent bien sûr que les machines distantes exécutent un serveur SSH et/ou un serveur VNC.

3.8.3. Caractéristiques de l'EIAH

VITELS est un EIAH modulaire fondé sur quatre composants : le serveur portail, le LMS WebCT, le serveur central d'authentification et d'autorisation (AAI), et le serveur de gestion de ressources qui gère la planification des séances (voir la partie gauche de la Figure 3-9). L'accès aux activités s'effectue via une page de cours du LMS WebCT, après que ce dernier ait vérifié l'identité de l'apprenant au niveau du serveur d'authentification ainsi que son droit d'accès au laboratoire.

3.8.4. Support à la conception pédagogique

VITELS est fondé sur la théorie constructiviste qui considère l'apprentissage comme un processus de construction de connaissances résultant de l'interaction avec l'environnement d'apprentissage. L'environnement est constitué du laboratoire (distant) et de ressources d'apprentissage structurées en quatre chapitres : présentation des objectifs du cours, introduction aux fondements théoriques sous-jacents et test d'évaluation des prérequis, réalisation de l'activité de télé-TP, et enfin évaluation de connaissances et compétences.

3.8.5. Support de l'apprentissage collaboratif et du tutorat en ligne

L'environnement VITELS étant fondé sur l'autonomie des apprenants, l'apprentissage collaboratif en ligne n'est supporté que par les outils de communication et de collaboration du LMS WebCT. De même pour le tutorat, le projet VITELS propose très peu d'outils restreints au forum, au courrier électronique ou au téléphone.

3.8.6. Synthèse

VITELS est un projet ambitieux de télé-TPs dans le domaine des réseaux et télécommunication qui vise non seulement à fournir l'accès aux laboratoires en ligne à un nombre de plus en plus important d'apprenants à travers la mutualisation des ressources de toutes les universités partenaires, mais aussi d'atteindre l'efficacité pédagogique des télé-TPs par la mise en œuvre des principes du constructivisme et de l'apprentissage par essais et erreurs.

Pour atteindre ces objectifs, les développeurs du projet ont opté d'un côté pour une structuration modulaire commune à tous les télé-TPs qui préparent l'apprenant à l'activité d'expérimentation, et d'un autre côté en permettant aux apprenants de revenir aux conditions initiales des expériences à travers un mécanisme sécurisé de réinitialisation des laboratoires distants.

Malheureusement, ce projet n'offre peu ou pas d'outil support à l'apprentissage collaboratif et au tutorat en ligne, alors que des sondages effectués auprès des apprenants ont confirmé la nécessité de ces deux aspects.

3.9. Le projet Tele-Lab IT-Security

3.9.1. Caractéristiques générales

Le projet Tele-Lab IT-Security vise la création d'un environnement d'apprentissage dans le domaine de la sécurité informatique en donnant aux apprenants universitaires la possibilité de mettre en pratique leurs connaissances à travers un laboratoire à distance fondé sur des machines virtuelles.

Le principe du projet est d'embarquer dans un serveur web un laboratoire sous la forme d'une *pool* de machines virtuelles, et permettre aux utilisateurs d'accéder à distance à ces machines. Un apprenant aura à sa disposition un vrai système Linux accessible via une applet Java intégrée dans le navigateur web. Cet applet repose sur un client VNC qui affiche la fenêtre du bureau de la machine (virtuelle) Linux. L'équipe du projet travaille en parallèle sur la production de cours multimédia, de scripts d'exercices interactifs, de logiciels de

démonstration et de simulations relatives au domaine de la sécurité informatique (Hu et al., 2004a).

3.9.2. Caractéristiques du laboratoire distant

Le projet Tele-Lab utilise des laboratoires émulés fondés sur la technologie de virtualisation. Plus précisément, il s'agit de machines virtuelles exécutées dans le serveur du laboratoire qui offrent un environnement réaliste pour l'expérimentation. Les bénéfices pédagogiques de cette technologie sont divers :

- Le réalisme des expériences pour l'acquisition de compétences professionnelles.
- La mise en œuvre de l'apprentissage par essais et erreurs grâce aux opérations d'arrêt, de démarrage et de réinitialisation à distance.
- La prise en compte d'un grand nombre d'apprenants à la fois, grâce à la multiplication des machines virtuelles sur le même système physique.
- La portabilité des machines virtuelles d'un système à un autre, voir même la possibilité d'importer toute la machine virtuelle chez l'apprenant ou l'enseignant.

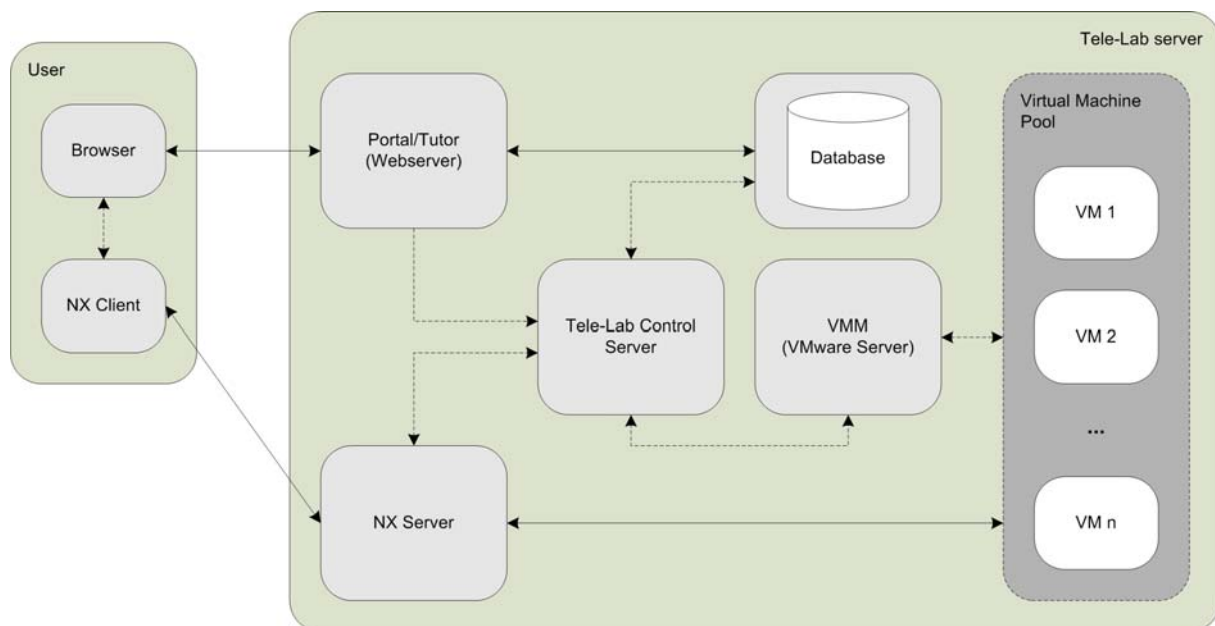


Figure 3-10. Architecture générale du projet Tele-Lab IT-Security. (Willems, 2008)

Comme l'illustre la Figure 3-10, le système Tele-Lab est embarqué dans un seul serveur physique. Cette architecture permet de construire rapidement un laboratoire virtuel à partir d'un support d'installation amovible, et par conséquent il est possible de répondre rapidement

à la montée en charge de la demande. Dans le serveur Tele-Lab, un nombre fixe de machines virtuelles est instancié dès le départ ce qui va simplifier la politique de réservation et d'allocation. Celles-ci peuvent être exploitées à distance par les moyens d'exploitation des machines physiques. L'accès en console sur les machines virtuelles est remplacé par un mécanisme de console virtuelle dans le cas où la machine virtuelle n'est pas accessible sur le réseau.

Au niveau de l'interface, l'instrumentation du laboratoire s'effectue via l'applet web « NX Client ». Il s'agit d'une applet qui affiche l'environnement graphique du bureau d'une machine distante (physique ou virtuelle) pour peu qu'elle soit contrôlée par un serveur NX.

3.9.3. Caractéristiques de l'EIAH

Tele-Lab est un EIAH qui intègre, en plus du laboratoire émulé, le serveur de cours et un tuteur intelligent en ligne. Chaque apprenant possède un profil qui décrit les étapes franchies et les difficultés rencontrées. A partir de ces informations, le tuteur intelligent construit et affiche un hypermédia personnalisé constitué de cours et ressources que doit consulter l'apprenant avant de passer aux pages des exercices qui donnent accès à la fenêtre d'expérimentation. La relation entre ces différents composants est illustrée dans la Figure 3-10.

Toutefois, bien qu'il soit fondé sur des technologies ouvertes en termes de contenus et de technologies de virtualisation, Tele-Lab est un système fermé où les ressources d'apprentissage sont fortement couplées aux machines virtuelles.

3.9.4. Support à la conception pédagogique

Le projet Tele-Lab est fondé sur la théorie d'apprentissage constructiviste à travers la mise en œuvre du principe de l'apprentissage par essais et erreurs, et en utilisant un système de tuteur intelligent de type hypermédia pour guider l'apprentissage (Cordel et al., 2008).

Nous n'avons pas trouvé de documentation sur la scénarisation des télé-TPs, bien que le tuteur intelligent soit programmé pour exécuter des scénarios en fonction des profils d'apprenants.

3.9.5. Support à l'apprentissage collaboratif en ligne et au tutorat en ligne

Vu ses fondements constructivistes, Tele-Lab ne supporte ni la collaboration, ni le tutorat en ligne. Télé-Lab n'est utilisé que comme support à l'enseignement en classe où le tutorat et la collaboration prennent place en présence.

3.9.6. Synthèse

Le projet Tele-Lab IT-Security est un EIAH spécifique à l'enseignement de l'informatique qui exploite les technologies nouvelles pour délivrer des télé-TPs réalistes en tentant de satisfaire la grande demande sur ces activités. C'est un EIAH qui convient à un enseignement mixte (présentiel et à distance) afin d'assurer la collaboration et le tutorat qui ne sont pas supportés par l'outil. Aussi, bien que l'outil soit innovant, il n'exploite pas toutes les possibilités offertes par les technologies de virtualisation comme l'implémentation d'expériences plus complexes.

3.10. Synthèse générale

Nous avons mené dans ce chapitre une étude fondée sur un modèle d'analyse issu des différents critères influant l'efficacité pédagogique des télé-TPs et identifiés dans le chapitre 2. Cette investigation, dont les détails figurent en Annexe A, concerne huit EIAHs dédiés aux télé-TPs, et nous a permis de mettre en exergue les points forts que nous pouvons reprendre, ainsi que les points faibles dont nous devons apporter des solutions pertinentes.

Les points forts constatés concernent essentiellement la mise en ligne de laboratoires réels ou émulsés, et les interfaces de télé-manipulation de ces laboratoires qui visent à donner un sentiment de réalisme et/ou de télé-présence. En effet, ces sentiments favorisent d'un côté l'acceptation des télé-TPs de la part des enseignants et apprenants, et d'un autre ils aident à acquérir des compétences pratiques et professionnelles réelles. Notons que par rapport à ce dernier point, un vide persiste et concerne la séparation des concepts de l'expérience (qui consiste en une configuration initiale visant des objectifs précis), du laboratoire (qui constitue une installation technique), et du télé-TP (qui est une activité d'apprentissage composée de diverses ressources).

Les points faibles concernent l'intégration des télé-TPs dans les EIAHs existants et la mise en œuvre de l'apprentissage collaboratif et du tutorat en ligne. Pour le premier aspect, il faudrait intégrer les télé-TPs et leurs composants (expériences et laboratoires) dans un processus de normalisation, seul moyen d'assurer l'interopérabilité entre ressources et plateformes d'apprentissage. Ceci aura comme conséquence de faciliter le travail de conception et de déploiement de télé-TPs pour les enseignants concepteurs qui vont intégrer dans les télé-TPs d'autres ressources d'apprentissage utiles.

Pour les deux autres points, nous devons concevoir et implémenter des outils et IHMs les plus génériques possibles, qui favorisent l'expérimentation collaborative et le suivi des apprenants dans ce contexte. Les domaines de l'apprentissage et du travail assisté par ordinateurs devraient y apporter des éléments de réponse pertinents.

Chapitre 4. Les verrous à lever

4.1.	<u>L'hétérogénéité des modèles de laboratoires</u>	98
4.2.	<u>L'hétérogénéité des moyens de télé-instrumentation</u>	98
4.3.	<u>La rigidité des modèles d'expériences</u>	98
4.4.	<u>La fédération et la mise en œuvre d'expériences distribuées</u>	99
4.5.	<u>L'hétérogénéité et l'inaccessibilité des traces d'activités</u>	100
4.6.	<u>L'intégration avec les normes et standards pédagogiques existants</u>	101
4.7.	<u>Support à l'expérimentation collaborative et au tutorat en ligne</u>	101
Chapitre 4.	Les verrous à lever	97
4.1.	L'hétérogénéité des modèles de laboratoires	98
4.2.	L'hétérogénéité des moyens de télé-instrumentation	98
4.3.	La rigidité des modèles d'expériences	98
4.4.	La fédération et la mise en œuvre d'expériences distribuées	99
4.5.	L'hétérogénéité et l'inaccessibilité des traces d'activités	100
4.6.	L'intégration avec les normes et standards pédagogiques existants	101
4.7.	Support à l'expérimentation collaborative et au tutorat en ligne	101
Partie II :	Une architecture et des modèles génériques pour l'intégration des télé-TPs en EIAHs.....	104

Le chapitre précédent a révélé que la recherche sur les EIAHs dédiés aux télé-TPs a bien avancé à travers les solutions pertinentes proposées. Mais nous avons également constaté que chacun des projets étudiés s'est focalisé sur un ou plusieurs aspects, et négligé d'autres propriétés tout aussi importantes. Par ailleurs, peu de projets ont apporté des solutions pérennes ou réutilisables, en particulier à cause de la disparité des modèles proposés pour les télé-TPs et leurs principaux composants : les laboratoires et les expériences.

Dans la suite de ce chapitre, nous dressons un ensemble de lacunes qui constituent, de notre point de vue, les verrous à lever pour faciliter la généralisation et l'efficacité des télé-TPs dans les plateformes de télé-enseignement existantes.

4.1. L'hétérogénéité des modèles de laboratoires

La modélisation des dispositifs de laboratoires distants est une condition primordiale à leur intégration et à leur interopérabilité avec les EIAHs existants. En effet, un laboratoire distant devrait être assimilé à une ressource pédagogique d'apprentissage au même titre que les cours, exercices et autres objets pédagogiques traditionnels.

La généricité des activités de télé-TPs dépend de la normalisation des modèles de laboratoires. Si la majorité des projets étudiés dans le chapitre précédent ont introduit des méta-modèles ouverts pour leurs laboratoires, tous ces travaux adoptent des approches différentes allant de la modélisation textuelle aux ontologies pour représenter les dispositifs distants, les activités, et les IHMs d'instrumentation correspondantes. Le manque d'un modèle standard, générique et unificateur qui puisse couvrir tous les équipements, dispositifs et logiciels de laboratoires de tout type, qu'ils soient réels, virtuels ou hybrides, est flagrant.

4.2. L'hétérogénéité des moyens de télé-instrumentation

Les moyens d'instrumentation sont également un frein à l'interopérabilité entre les EIAHs existants et les dispositifs de laboratoires. En effet, différents protocoles de communication existent, mais ils sont fondés sur des technologies d'instrumentation conçues par divers constructeurs. Par ailleurs, de plus en plus de dispositifs disposent de leurs propres systèmes d'exploitation informatiques embarqués dont il faudrait tenir compte pour une exploitation avancée et optimisée des instruments.

Enfin, il manque un protocole de communication normalisé et une interface de commande unificatrice qui enveloppent les protocoles et interfaces existants afin de : (1) faciliter le développement d'IHMs de télé-instrumentation interopérables avec les différentes plateformes d'expérimentation, (2) assurer le partage des ressources de télé-TPs entre ces différentes plateformes, et (3) assurer la communication entre ces plateformes.

4.3. La rigidité des modèles d'expériences

La plupart des projets étudiés associent les expériences aux dispositifs de laboratoires, alors que nous avons montré leurs différences d'un point de vue pédagogique mais également technique dans la section 2.2. Comme pour les laboratoires, il manque un méta-modèle pour

décrire les expériences qui sont caractérisées par les conditions particulières de leurs implémentations.

Un tel méta-modèle doit prendre en considération aussi bien les différents composants qu'une expérience peut utiliser selon le domaine d'enseignement visé et les technologies choisies (instruments, dispositifs, logiciels, etc.), les liens entre ces composants (comme dans le cas d'une topologie de réseau), mais aussi leur configuration initiale. Dans le cas d'expériences réparties sur plusieurs laboratoires délocalisés, le méta-modèle doit fournir les moyens d'exprimer ce caractère distribué.

Enfin, afin de faciliter le développement d'IHMs de type tableau de bord, le modèle d'une expérience doit être continuellement mis à jour par les variables qui en décrivent l'état. Ces dernières sont affectées par les réactions de l'expérience aux actions des utilisateurs et par les événements qui y surviennent.

4.4. La fédération et la mise en œuvre d'expériences distribuées

En conséquence aux trois problématiques évoquées plus haut, il n'est pas possible de fédérer les différentes ressources et infrastructures de laboratoires, mises à disposition par différentes institutions ou organisations au service de l'expérimentation pédagogique et scientifique. Ce concept d'expériences collaboratives est similaire aux « Collaboratoires » (Agarwal et Johnston, 1998), où des chercheurs appartenant à différents laboratoires collaborent à la résolution de problèmes complexes.

La fédération des ressources vise à mutualiser les dispositifs détenus par plusieurs organismes de formation partenaires dans le but de minimiser les coûts d'acquisition et d'exploitation, mais aussi d'offrir plus de temps d'expérimentation au plus grand nombre d'apprenants dans les meilleures conditions d'apprentissage et de manière équitable. Elle vise également l'implémentation d'expériences virtuelles distribuées sur plusieurs laboratoires, à l'image des expériences sur les bancs d'essais et aux calculs réalisés par des grilles. En plus de l'homogénéisation des méta-modèles de description des différentes ressources (laboratoires et expériences) et de l'harmonisation des protocoles et services de télé-instrumentation, la fédération de laboratoires distants nécessite un système d'authentification et d'autorisation des différents utilisateurs couplé à un système de réservation de ressources. Un système

d'informations relatif aux utilisateurs, à leurs rôles et privilèges, ainsi qu'à l'état d'allocation des ressources, doit être mis en œuvre. Nous pouvons nous inspirer des solutions implémentées par les systèmes de gestion de grilles de calcul, comme le système WMS de l'infrastructure GLITE²⁹, pour résoudre ce problème. En effet, une base d'informations sur l'état de disponibilité et d'allocation des ressources de calcul et de stockage est constituée et continuellement mise à jour. Fondée sur un modèle de description des ressources, ce système d'information est hiérarchisé en trois niveaux : le niveau de la ressource, le niveau de l'institution, le niveau de l'organisation virtuelle (constitué par un ensemble d'institutions partenaires). Ainsi, chaque niveau alimente le niveau supérieur en informations relatives à l'état des ressources qu'il gère (Cecchi et al, 2009).

4.5. L'hétérogénéité et l'inaccessibilité des traces d'activités

Certains projets comme PEARL et eINST proposent des mécanismes évolués pour l'enregistrement et l'exploitation des traces d'activités. La gestion et l'exploitation de ces informations sont une problématique récurrente dans le domaine des EIAHs, du fait que les traces permettent d'analyser les usages, et supporter certaines fonctions pédagogiques comme le tutorat et l'apprentissage collaboratif. Plus généralement, les traces informatiques ont toujours posé des problèmes aux administrateurs du fait de leur hétérogénéité qui rend difficile le développement d'applications génériques d'analyse et de corrélation d'événements.

Un modèle de structuration standard et unificateur pour les traces de télé-TPs, ainsi que des mécanismes associés, doivent rendre possibles les opérations de filtrage, d'enregistrement et de recherche de ces traces afin de supporter les fonctions pédagogiques de tutorat, de travail collaboratif et d'analyse a posteriori des télé-TPs. D'autre part, nous sommes convaincus que ces informations doivent être externalisées de la plateforme d'apprentissage ou du laboratoire distant, afin qu'elles puissent être réutilisées par différents environnements visant la personnalisation des services offerts aux utilisateurs.

²⁹ <http://glite.cern.ch/glite-WMS/>

4.6. L'intégration avec les normes et standards pédagogiques existants

Nous avons évoqué dans le chapitre 2 la tendance actuelle à normaliser les ressources d'apprentissage pour des raisons d'interopérabilité techniques, mais également pour des considérations pédagogiques et économiques liées à la réutilisation, au partage et à l'échange de ces ressources.

Des travaux de recherche ont démontré que les normes actuelles d'intégration des objets pédagogiques comme IMS-CP ou SCORM, ne couvrent pas la totalité des services requis pour l'interaction avec des ressources hautement interactives telles que les expériences en laboratoires (Buitrago, 1998 ; Guéraud et al, 2004 ; Guéraud, 2005), et proposent que ce genre d'objets offrent quatre services (Guéraud et al., 2004) : «

*La référence : l'OPI³⁰ peut fournir la liste de ses variables ;
l'Inspectabilité : l'OPI peut délivrer, à la demande, ses propriétés ;
L'observabilité : l'OPI peut signaler l'occurrence d'événements ;
La scriptabilité : l'OPI permet de modifier son état. »*

Grâce à ces services, l'intégration totale des expériences au sein des scénarios d'apprentissage conformes au standard IMS-LD ou LDL devient envisageable.

Aussi, l'adaptation au standard LOM est nécessaire pour décrire efficacement les expériences afin de les intégrer dans les viviers d'objets pédagogiques en vue de leur partage et réutilisation.

4.7. Support à l'expérimentation collaborative et au tutorat en ligne

Beaucoup de travaux se sont intéressés aux problématiques de l'apprentissage collaboratif et du tutorat en ligne pour vaincre l'isolement et améliorer l'apprentissage en ligne à la lumière des nouvelles théories d'apprentissage socioconstructivistes et situées. ; de plus, les chapitres précédents ont démontré l'importance de ces deux aspects dans le cadre des activités de travaux pratiques traditionnels. Cependant, un manque demeure béant au niveau des outils dédiés à l'expérimentation collaborative et au tutorat en ligne en temps réel pour les activités

³⁰ Object Pédagogique Interactif

de télé-TPs. Pourtant, lors de la participation à de telles activités, les apprenants ont souvent besoin d'une aide rapide et ponctuelle pour surmonter une situation de blocage sous-peine d'être démotivés rapidement. La problématique ici peut être divisée en deux parties :

- Dans quelles mesures les résultats des travaux sur la collaboration et le tutorat, appliqués aux enseignements traditionnels en ligne, peuvent être réutilisés ou ré-implémentés pour les télé-TPs ?
- Quels outils et IHMs sont nécessaires au tutorat et à l'expérimentation collaborative en ligne, en temps réel ou en différé, et quelles sont leurs modalités d'utilisation ?

Cette problématique peut également s'exprimer ainsi : comment intégrer les télé-TPs aux EIAHs, tout en articulant les outils supports à l'expérimentation en ligne avec les outils de collaboration et de tutorat existants ? Dans la revue de littérature du chapitre 3, nous avons identifié quelques éléments de réponse à travers les aspects de présence sociale (ou psychologique) et de conscience mutuelle (ou *Awareness*) qui renforcent le réalisme des (télé)manipulations sur des laboratoires distants. La présence dépend des propriétés de l'IHM de télé-TP qui doit procurer aux apprenants le sentiment de présence virtuelle dans une communauté distribuée pour tenter de remplacer partiellement le sentiment de présence physique dans un laboratoire réel. La conscience mutuelle, quant à elle, se rattache aux caractéristiques de l'IHM de télé-TP qui permettent à chaque membre d'une équipe virtuelle de manipuler les mêmes objets informatiques (correspondants à des objets physiques, émulés ou simulés), tout en percevant en temps réel les actions effectuées par les autres membres (y compris le tuteur) sur ces objets. Nous pensons que la mise à disposition aux différents acteurs des télé-TPs des traces d'activités effectuées individuellement ou collectivement sur l'expérience en temps réel ou en différé, devrait accroître le sentiment de présence sociale et de conscience mutuelle. Les travaux de notre équipe apportent des solutions à ce problème à travers un méta-modèle générique de représentation des traces et d'un ensemble d'interfaces normalisées de recueil, de stockage, d'externalisation et d'exploitation de ces traces (Broisin et Vidal, 2007).

Partie II : Une architecture et des modèles
génériques pour l'intégration des télé-TPs en
EIAHs

Chapitre 5. Une architecture pour l'intégration des EIAHs et des environnements de télé-TPs

5.1.	<u>Une architecture structurée en couches</u>	105
5.2.	<u>La couche d'Apprentissage</u>	108
5.2.1.	<u>L'objet pédagogique « Expérience »</u>	110
5.2.2.	<u>Le moteur d'exécution des objets « Expérience »</u>	115
5.2.3.	<u>L'IHM de télé-opération</u>	117
5.3.	<u>La couche d'Expérimentation</u>	125
5.3.1.	<u>Architecture des Pilotes de dispositifs</u>	128
5.3.2.	<u>Le protocole de communication avec la couche supérieure</u>	129
5.3.3.	<u>Une interface standardisée et unificatrice</u>	129
5.3.4.	<u>La problématique du déploiement automatique des expériences</u>	130
5.4.	<u>Synthèse</u>	132

Le chapitre précédent a mis en exergue des verrous relatifs à l'hétérogénéité des modèles de laboratoires, d'expériences, et de traces d'activité. Pour pallier ces manques, nous proposons dans cette partie une solution s'articulant autour de deux éléments clés : (1) une architecture structurée en couches assurant une communication transparente des contextes d'apprentissage (EIAH) et d'expérimentation (laboratoires en ligne), et (2) un méta-modèle qui, d'une part, décrit de façon formelle les éléments spécifiques d'un télé-TP (expériences et laboratoires) et les activités réalisées par les utilisateurs sur ces entités, et d'autre part dirige le moteur d'exécution qui contrôle et supervise les expériences et laboratoires. Ce chapitre se focalise sur l'architecture de notre système et se concentre en particulier sur ses couches supérieure et inférieure ; le chapitre suivant détaille la couche centrale.

5.1. Une architecture structurée en couches

La Figure 5-1 illustre l'architecture globale de notre proposition, articulée autour de trois couches :

- La couche d'Apprentissage représente l'interface de tout le système avec les acteurs humains. Elle correspond aux environnements informatiques pour l'apprentissage humain existants, étendus par des interfaces et composants dédiés à la réalisation d'activités de travaux pratiques à distance.
- La couche d'Intégration renferme l'ensemble des composants nécessaires à la mise en œuvre de la communication transparente entre les EIAHs de la couche d'Apprentissage et les laboratoires en ligne compris dans la couche d'Expérimentation. Le niveau supérieur de cette couche, exposé aux outils additionnels intégrés dans les EIAHs, propose un ensemble de services dédiés au déploiement, à l'exploitation et à la gestion des expériences en ligne ; il expose également d'autres services dédiés à la gestion des traces d'activités réalisées par les acteurs des télé-TPs. Le cœur de la couche d'Intégration comprend une base de données renfermant des informations décrivant l'ensemble des entités de notre architecture globale, ainsi qu'un composant assurant la communication avec les laboratoires distants compris dans la couche d'Expérimentation ; cette entité permet d'opérationnaliser les services exposés aux utilisateurs.
- La couche d'Expérimentation représente l'environnement du laboratoire en ligne, et se compose d'un ensemble de laboratoires qui peuvent être distribués sur plusieurs sites, chacun comprenant un ou plusieurs dispositifs techniques physiques ou virtuels. Ces dispositifs sont munis d'éléments de pilotage appelés « Pilote » sur la Figure 5-1 permettant le pilotage et l'observation à distance des laboratoires.

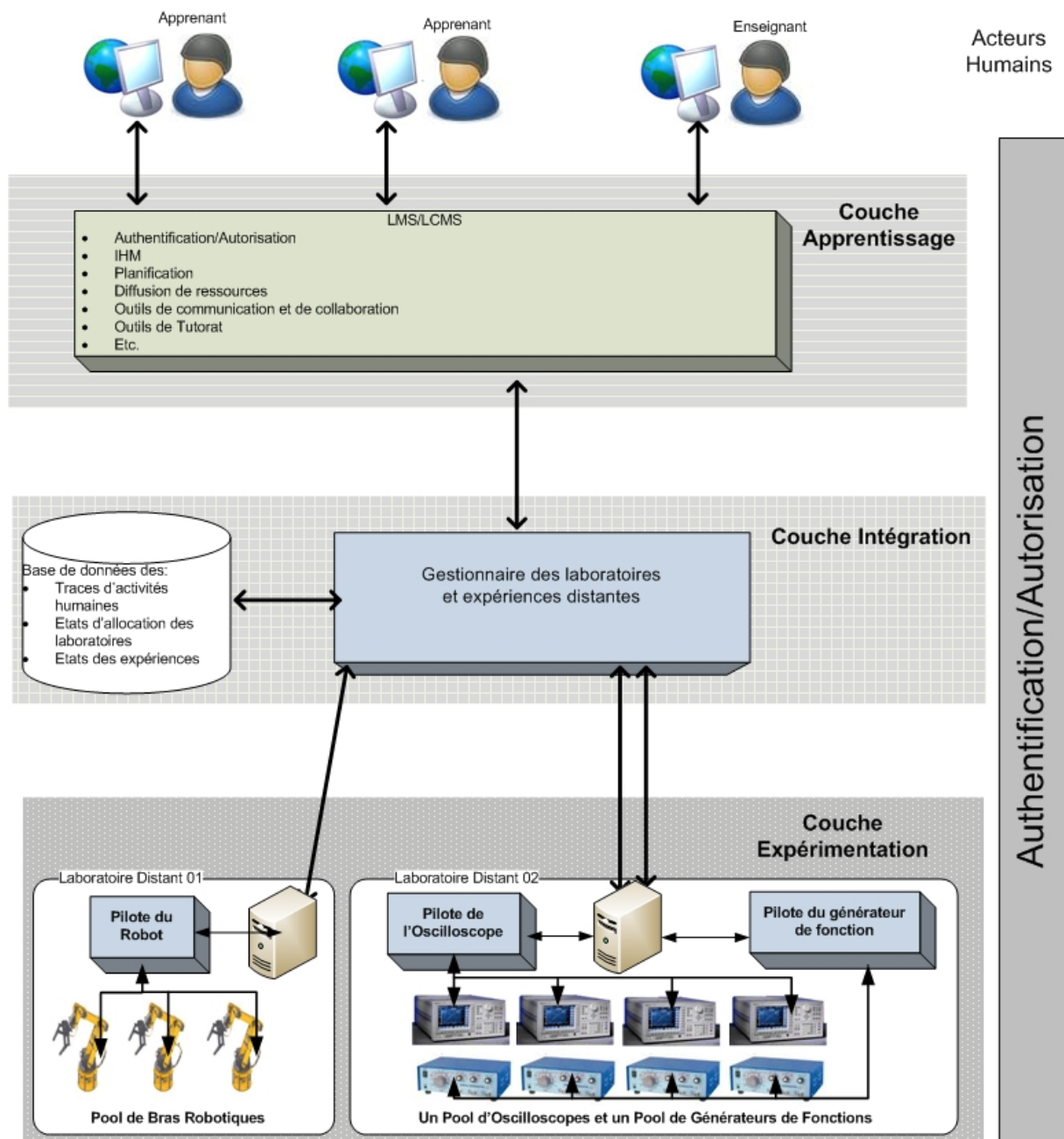


Figure 5-1. Une architecture pour l'intégration des EIAHs et des télé-TPs.

Dans la suite de ce chapitre, nous nous focalisons sur les couches supérieure et inférieure de l'architecture proposée. Nous en détaillons les composants, IHMs et fonctionnalités offertes, alors que les deux prochains chapitres seront dédiés à la couche d'Intégration et aux différents modèles supports de notre approche.

5.2. La couche d'Apprentissage

L'intégration des activités de travaux pratiques dans les EIAHs existants ne peut être efficace sans la prise en compte des normes et standards relatifs aux objets pédagogiques et aux activités d'apprentissage. Leur pertinence d'un point de vue pédagogique fait l'objet de débats et de controverses continus (Salmi et Jaillet, 2005), mais ces normes et standards constituent les bases de la plupart des plateformes d'apprentissage existantes. Ces dernières doivent alors être pourvues du ou des composants informatiques appropriés pour l'intégration et l'exploitation des différentes ressources pédagogiques comprises dans un scénario d'apprentissage.

Dans un objectif d'homogénéisation des différents objets pédagogiques, notre approche consiste à considérer les télé-TPs comme des activités d'apprentissage ordinaires qui intègrent des objets pédagogiques spécifiques : les « Expériences ». Ces dernières nécessitent un moteur d'exécution qui en assure le pilotage aussi bien par les humains que par les applications informatiques de contrôle pédagogique comme les IHMs et les moteurs de scénarii IMS-LD. La Figure 5-2 illustre une expérience en ligne dans le domaine du génie électrique, et présente les composants du moteur d'exécution que nous proposons d'introduire dans la couche d'Apprentissage, et en particulier au sein des plateformes pédagogiques existantes, pour exploiter les objets « Expérience » :

- *L'Agent de pilotage* assure l'interaction à distance avec les expériences en ligne. C'est l'élément avec lequel interagit l'IHM de télémanipulation, le moteur de scénarii IMS-LD ou toute autre application de contrôle pédagogique. Cet agent permet la manipulation de l'ensemble des dispositifs de chaque laboratoire, mais il assure également leur supervision en retrouvant, via la couche d'Intégration, l'état de ces dispositifs.
- *L'Agent de gestion des traces d'activités* recueille, stocke et retrouve toutes les activités effectuées par les acteurs humains sur les expériences en ligne. Ce composant est une brique essentielle pour les services et outils supports au tutorat et à l'apprentissage collaboratif en ligne, ainsi qu'aux outils d'analyse et de supervision des comportements humains face aux expériences en ligne.

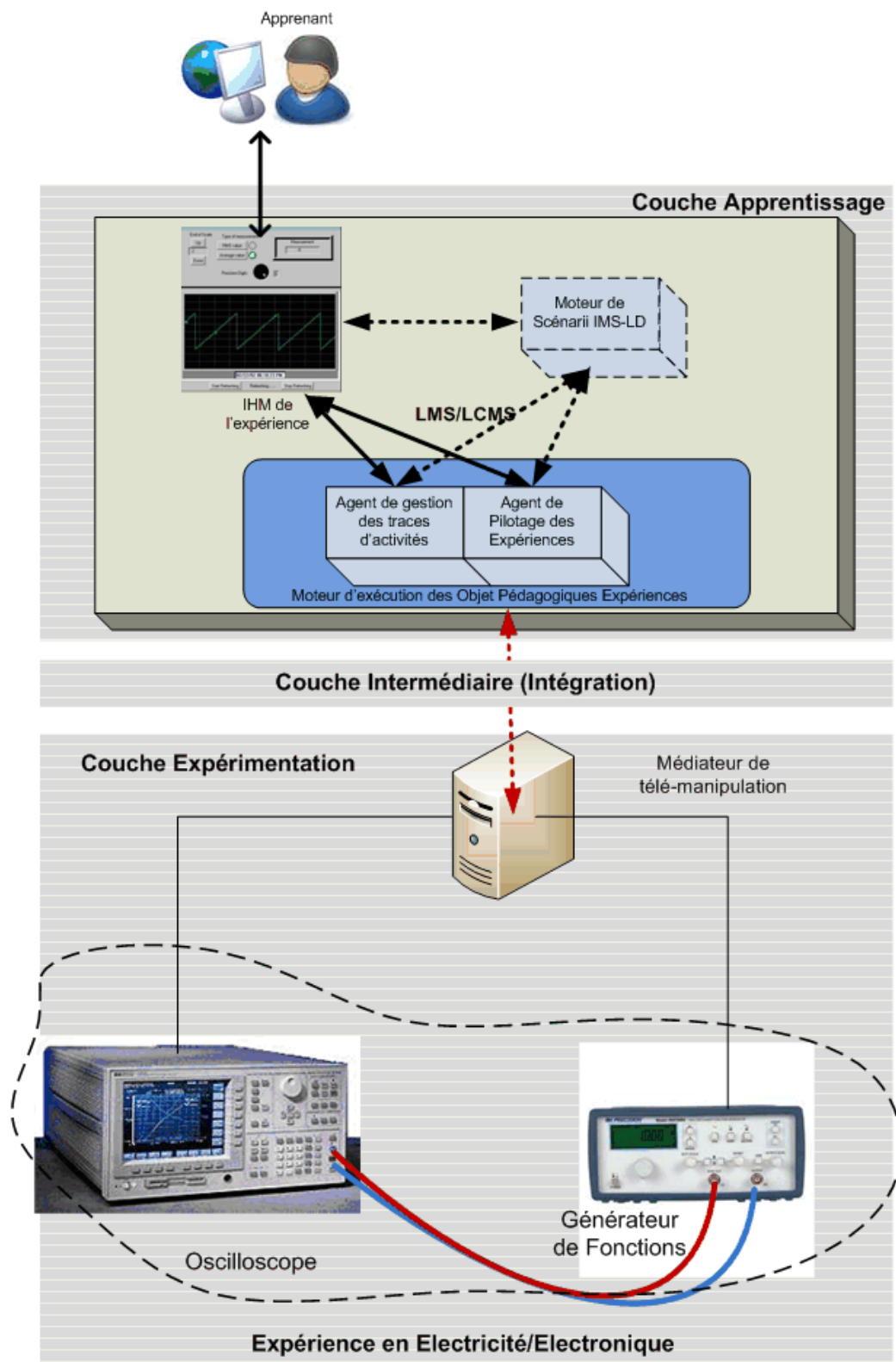


Figure 5-2. Détails de la couche d'Apprentissage.

Pour exploiter le moteur d'exécution, nous proposons une IHM intégrée à l'EIAH existant qui permet aux différents acteurs humains (apprenants, enseignants et tuteurs) de télé-opérer des objets « Expérience » ; elle comprend divers outils pédagogiques, ainsi qu'un composant dédié à l'observation et à la réutilisation des traces d'activités réalisées sur les expériences en ligne.

Avant de rentrer dans les détails de notre moteur d'exécution et de l'IHM exploitant les fonctionnalités qu'il propose, nous présentons l'objet pédagogique « Expérience ».

5.2.1. L'objet pédagogique « Expérience »

Nous introduisons ici l'objet pédagogique « Expérience » qui formalise le concept de télé-TP défini et caractérisé dans le chapitre 2 de ce manuscrit. Une « Expérience » suit alors le même cycle de vie que tout autre objet pédagogique (IEEE, 2003) : (1) la conception, (2) le déploiement, (3) l'exploitation, (4) la suspension, (5) la sauvegarde, et (6) la restauration.

5.2.1.1. Cycle de vie de l'objet pédagogique « Expérience »

5.2.1.1.1. La phase de conception

La conception représente la phase clé de tout le cycle de vie d'une expérience car elle doit traduire les objectifs pédagogiques visés par l'enseignant concepteur. La conception s'effectue généralement (voir chapitre 3) par l'édition d'un fichier texte conforme à un langage spécifique tel que XML pour décrire les différents objets composant l'expérience ainsi que leurs attributs, méthodes et relations entre eux. Ce processus nécessitant des compétences supplémentaires de la part de l'enseignant concepteur, il est préférable d'offrir des IHMs intuitives permettant de construire aisément des expériences simples ou complexes. Ainsi, lorsqu'un enseignant valide la conception d'une expérience à travers l'éditeur graphique, l'IHM génère le modèle descriptif correspondant et conforme à la syntaxe et à la sémantique du langage de modélisation adopté.

Pour pallier aux insuffisances des méta-modèles de description (et de supervision) de laboratoires et d'expériences en ligne constatées dans le chapitre 4, notre approche repose sur un méta-modèle standard dédié à la gestion de toute entité logique ou physique. Fondé sur le

paradigme objet, un des avantages de ce méta-modèle présenté en détail dans le chapitre 7, est qu'il peut être exprimé selon divers langages :

- MOF (*Managed Object Format*) est le langage natif du méta-modèle. Les outils qui implémentent les concepts de ce méta-modèle standard s'appuient sur ce format.
- UML est destiné aux informaticiens capables de manipuler ce langage afin de faciliter la génération automatique de fichiers MOF.
- XML et OWL (*Web Ontology Language*) facilitent la manipulation du méta-modèle par les applications orientées web à travers le réseau.

Ainsi, un même modèle qui décrit une ou plusieurs entités (ainsi que leurs relations) peut être exploité de manière transversale aussi bien par la couche d'Apprentissage (via les formats XML ou OWL) que par les couches d'Intégration et d'Expérimentation (en utilisant le format MOF), et ce sans perte de sémantique. Dans notre contexte, l'IHM responsable de la génération du modèle correspondant à l'expérience conçue est décrite dans la section 5.2.3.

5.2.1.1.2. La phase de déploiement

Une fois l'expérience conçue, elle doit être déployée sur les laboratoires distants afin d'être exploitée par les apprenants et enseignants-tuteurs. Cette opération s'effectue donc lorsque l'activité de télé-TP est « ouverte » durant les plages horaires prédéfinies par l'équipe pédagogique. Le déploiement d'une expérience s'effectue en deux étapes :

- La réservation des ressources nécessaires à l'expérience (dispositifs physiques ou virtuels) auprès du ou des laboratoires mis à disposition pour la période de temps durant laquelle l'activité est ouverte.
- La configuration initiale des ressources réservées ainsi que la mise en place d'éventuelles interconnexions entre elles.

Ces deux étapes introduisent deux défis techniques et technologiques à lever :

- Le premier concerne la gestion d'une manière efficace, optimale et équitable des ressources mises à disposition par les laboratoires mutualisés afin de soutenir des politiques d'utilisation définies au niveau administratif.
- Le second concerne la configuration automatique des composants des expériences sur l'infrastructure de laboratoires distants.

La résolution des deux derniers problèmes d'une manière générique dépasse le cadre de ce travail, même si nous proposons dans la partie III des solutions applicables au domaine de l'enseignement de l'informatique. Nous présentons également dans la section 5.3.4 une solution adaptée au domaine du génie électrique (et tirée de la littérature) afin de démontrer la généralité de notre approche.

5.2.1.1.3. La phase d'exploitation

L'exploitation d'une expérience consiste à exécuter des opérations sur les dispositifs qu'elle contient, ainsi qu'à recevoir les messages de réponse ou d'événements survenus sur ces dispositifs. Les commandes/réponses/événements traversent toute la chaîne de notre architecture, en partant de l'IHM intégrée dans la couche d'Apprentissage pour arriver à la ressource télé-manipulée, en passant par la couche d'Intégration. Notons qu'afin d'assurer le suivi de l'apprentissage, ces opérations sont tracées à travers des mécanismes de recueil, de stockage et de consultation afin de supporter les diverses activités pédagogiques en ligne ou a posteriori que sont le travail d'équipe, le tutorat en ligne et la réingénierie de scénarii pédagogiques.

5.2.1.1.4. La phase de sauvegarde et de suspension

La sauvegarde d'une expérience consiste à enregistrer le modèle correspondant (constitué des différentes entités de l'expérience et des relations entre elles) pour pouvoir la restaurer par la suite dans cet état, alors que la phase de suspension consiste généralement à effectuer les opérations inverses de la phase de déploiement (libérer les ressources et les réinitialiser avec leurs paramètres d'origine).

5.2.1.1.5. La phase de restauration

La phase de restauration est similaire à la phase de déploiement, outre le fait que l'initialisation de l'expérience est fonction de son état lors de l'opération de sauvegarde. Cette fonction permet le prolongement d'une session d'expérimentation sur plusieurs séances par exemple.

Le modèle formel de l'objet pédagogique « Expérience » qui résulte de la phase de conception n'est exploitable dans un environnement d'apprentissage que s'il est enrobé d'informations

additionnelles pédagogiques. En effet, grâce à ces données supplémentaires, l'expérience devient un objet pédagogique stocké et indexé dans une banque d'objets pédagogiques, puis repérable et exploitable par les environnements informatiques pour l'apprentissage humain. Dans la section suivante nous exposons notre proposition fondée sur un standard, largement adopté, de description des objets pédagogiques.

5.2.1.2. Description des expériences

De nombreux standards existent aujourd'hui pour faciliter l'échange, le partage et la mutualisation des ressources pédagogiques. Trois types de standards pédagogiques doivent être distingués : les standards liés à la description et à l'indexation des objets pédagogiques comme *LOM (Learning Object Metadata – IEEE, 2009)*, les standards liés à l'intégration, l'agrégation et l'exécution d'objets pédagogiques comme *IMS Content Package* et *SCORM (Sharable Content Object Reference Model – ADLNet, 2009)*, et enfin les standards liés à la scénarisation pédagogique comme *IMS LD (Learning Design – IMSGlobal, 2003)*.

Dans notre approche, un objet pédagogique « Expérience » est décrit par une fiche de métadonnées LOM contenant une référence vers le modèle correspondant à l'expérience ; le choix de ce standard se justifie par son implémentation dans la plupart des systèmes de stockage de ressources d'apprentissage, ou viviers de connaissances (GLOBE, 2011), et facilite ainsi son indexation, son partage et sa réutilisation par les plateformes d'enseignement.

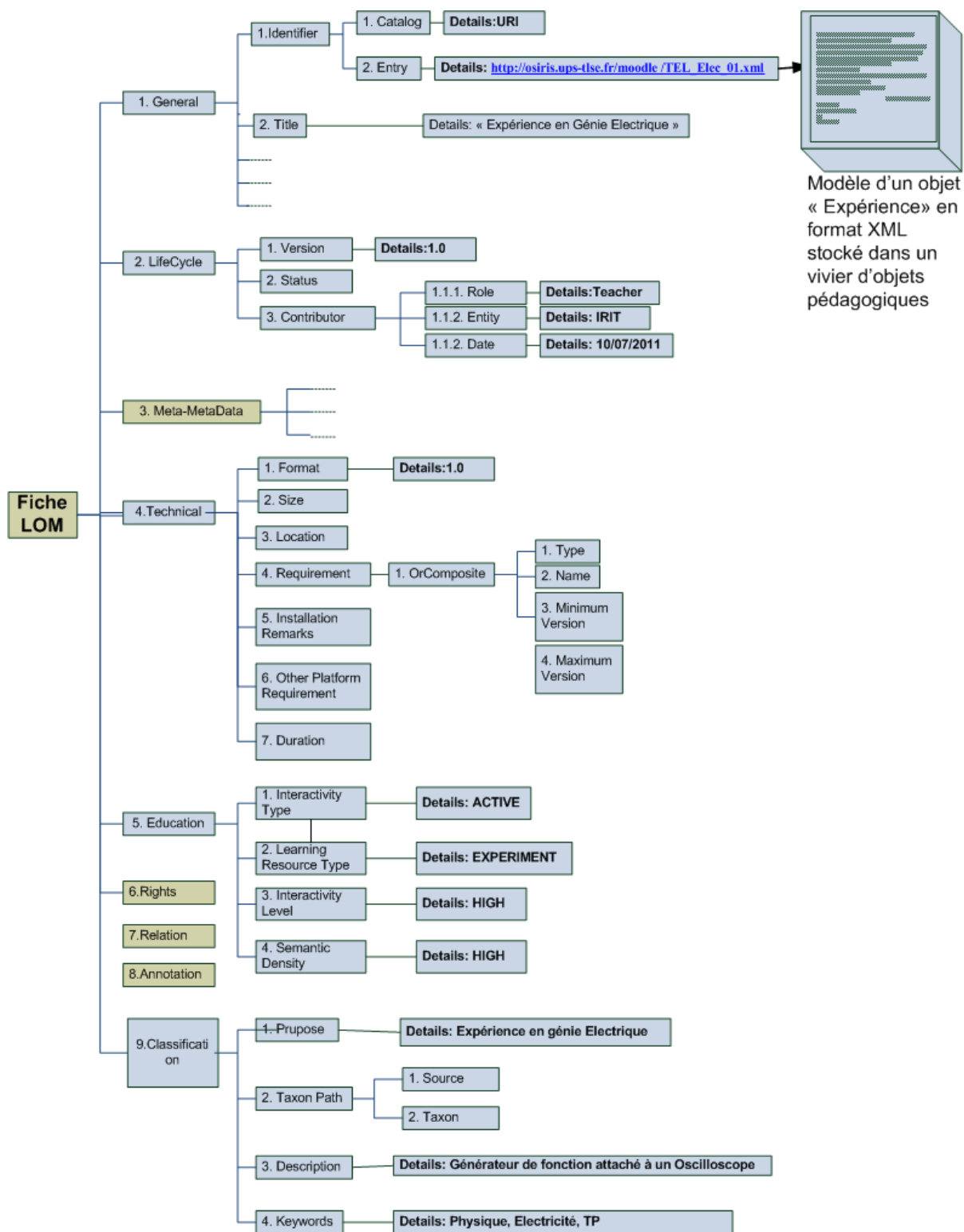


Figure 5-3. Contenu de l'objet pédagogique « Expérience ».

Ainsi, une expérience type en génie électrique pourrait être décrite par la fiche de métadonnées LOM suivante illustrée par la Figure 5-3 :

- Les champs 1.1.1 et 1.1.2 contiennent la localisation du fichier qui contient le modèle de l'expérience. Par exemple, si le champ « *Catalogue* » 1.1.1 vaut « *URI* », la valeur du champ « *Entrée* » 1.1.2 peut être http://osiris.ups-tlse.fr/moodle/TEL_Elec_01.xml.
- Les champs de la catégorie 2.X « *Cycle de vie* » doivent refléter les informations de version et de date de publication de l'objet, ainsi que celles liées aux contributeurs.
- Les champs de la catégorie 4.X « *Technique* » doivent renseigner les prérequis techniques à l'exploitation de l'expérience.
- Dans la catégorie 5.X « *Pédagogique* », nous proposons d'affecter la valeur « *Expérience* » au champ 5.2 « *Type de ressource* », la valeur « *Actif* » au champ 5.1 « *Type d'interactivité* », et la valeur « *Hautement interactif* » au champ 5.3 « *Niveau d'interactivité* ».

Pour qu'ils soient exploitables dans un EIAH, les objets pédagogiques « *Expérience* » doivent être manipulés par un composant logiciel spécifique qui, à partir du modèle de l'expérience, est capable d'exécuter les différentes opérations liées à son cycle de vie ; notre moteur d'exécution est présenté dans la prochaine section.

5.2.2. Le moteur d'exécution des objets « *Expérience* »

Le moteur d'exécution des expériences est composé de deux agents : l'agent de pilotage des expériences et l'agent de gestion des traces d'activités. Ce moteur, transparent aux utilisateurs finaux, interagit avec l'IHM de télé-opération d'un côté, et communique avec la couche d'Intégration via les technologies de l'Internet de l'autre.

5.2.2.1. L'agent de pilotage des expériences

Cet agent représente le composant intermédiaire entre les outils pédagogiques et les expériences en rendant possible la télé-opération d'un objet « *Expérience* ». Il reçoit de l'EIAH les instructions ordonnées par l'utilisateur, puis les renvoie à la couche d'Intégration pour qu'elles soient exécutées sur l'expérience distante. Dans l'autre sens, ce composant reçoit les messages issus de l'expérience en réponse aux actions exécutées et qui sont transmis par la couche d'Intégration.

Ainsi, l'agent de pilotage des expériences offre différents services aux applications pédagogiques pour surveiller et piloter une expérience. Il est pourvu de deux fonctions

principales : la surveillance par inspection (délivrance à la demande d'informations sur l'état courant de l'expérience) et notification (signalement de l'occurrence d'évènements survenus sur l'expérience) qui vise des objectifs d'observation, et la *scriptabilité* qui permet de modifier l'état d'une expérience. Ce composant n'agit pas directement sur l'expérience, il délègue l'opérationnalisation de ses fonctions aux services de la couche d'Intégration (voir chapitre 6). Il ne se soucie donc pas des détails techniques et spécifiques de la couche d'Expérimentation, et ce mécanisme de délégation garantit la transparence de l'intégration des expériences en libérant la couche d'Apprentissage de la gestion des aspects de bas niveau.

Les interfaces proposées par ce composant aux applications dédiées aux télé-TPs correspondent à celles des services exposés par la couche d'Intégration et qui sont détaillées dans le chapitre 6.

5.2.2.2. L'agent de gestion des traces d'activités

L'agent de gestion des traces d'activités assure deux tâches principales : il capture les traces informatiques correspondant aux activités réalisées par les utilisateurs dans l'IHM de télé-opération afin de les transmettre à la couche d'Intégration dans un objectif de stockage, et offre l'opportunité aux EIAHs, et notamment à l'IHM de télé-opération, de retrouver et de réutiliser ces traces en temps réel ou a posteriori dans un objectif de personnalisation des activités de télé-TP, de tutorat, etc. Cet agent capture alors, de manière transparente, trois types de traces qui seront détaillées dans le chapitre 7 :

- Les commandes émises par les acteurs humains.
- Les messages de réponses de l'expérience aux commandes des utilisateurs.
- Les messages d'évènements ou de notifications générés par les expériences.

Tout comme l'agent de pilotage des expériences, l'agent de gestion des traces s'appuie sur d'autres services offerts par la couche d'Intégration pour stocker et retrouver ces informations dans une base de données centralisée.

Ainsi, l'objet pédagogique « Expérience », associé aux composants du moteur d'exécution, correspond à un *Objet Pédagogique Interactif* (OPI) selon les termes de la définition de (Guéraud et al., 2004). En effet, notre ressource pédagogique offre un ensemble de services aux applications de contrôle pour être pilotée et surveillée : la Référence, l'Observabilité,

l'Inspectabilité et la Scriptabilité. Nous verrons dans le chapitre suivant comment ces services sont implémentés par la couche d'Intégration.

5.2.3. L'IHM de télé-opération

Notre interface dédiée à l'expérimentation en ligne s'appuie sur un certain nombre de règles et de principes généraux mis en avant dans le développement d'IHMs orientées web. L'objectif de cette IHM est de rendre transparent, pour l'utilisateur, l'accès au moteur d'exécution ; c'est à travers cette IHM que les enseignants et apprenants manipuleront un objet « Expérience » durant les diverses phases de son cycle de vie.

Nous insistons sur le fait que les interfaces présentées ici sont intégrées dans une plateforme d'apprentissage existante, elles ne constituent pas de nouveaux systèmes à part entière.

5.2.3.1. L'interface de conception

Nous proposons une IHM générique de conception divisée en deux parties et illustrée par la Figure 5-4 :

- La palette (à gauche) contient l'ensemble des éléments graphiques correspondant aux dispositifs de laboratoires supportés par la couche d'Expérimentation ; la présence de l'icône d'un dispositif dans la palette signifie que ce dispositif est enregistré au niveau de la couche d'Intégration, et que le Pilote qui en permet le contrôle et la supervision est déployé au niveau de la couche d'Expérimentation. La liste des dispositifs de laboratoires supportés est donc obtenue par interrogation de la couche d'Intégration qui retourne la liste des modèles associés aux dispositifs de laboratoires.
- La fenêtre centrale, ou console de travail, est l'espace où le concepteur effectue des opérations Glisser/Déposer à partir de la palette pour bâtir la maquette ou le gabarit de son expérience. L'outil offre également l'opportunité d'établir des connexions entre les composants.

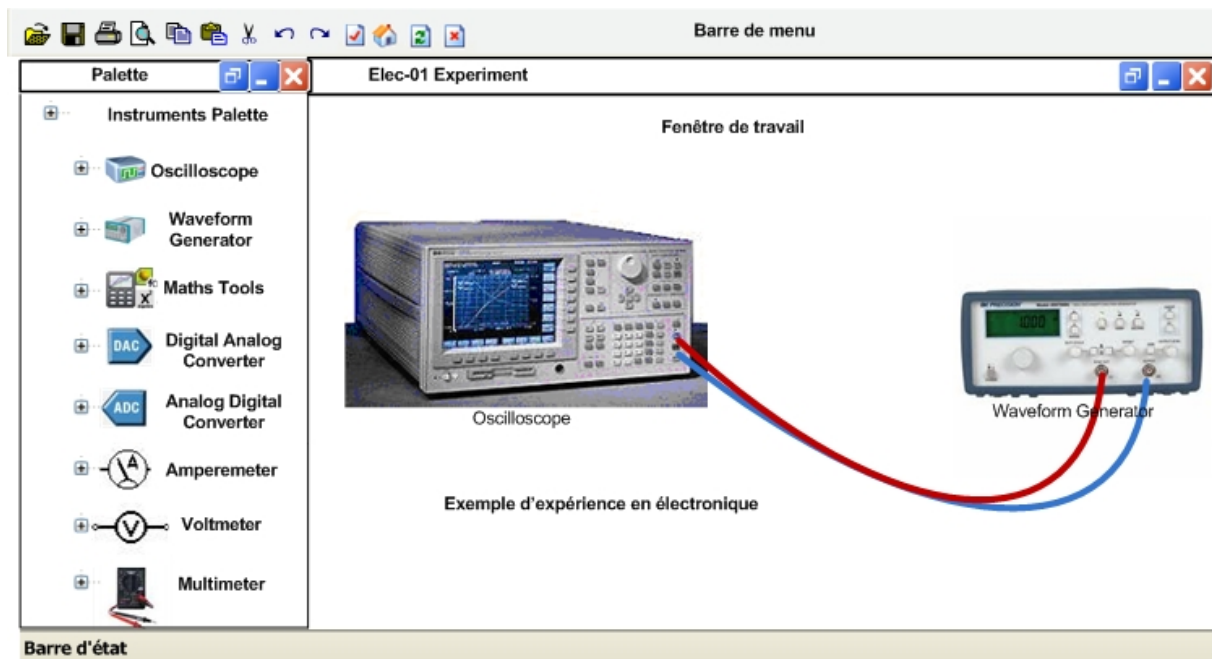


Figure 5-4. Prototypé d'IHM de conception d'une expérience en génie électrique.

La conception d'une expérience par cet outil auteur se traduit par la génération automatique du modèle correspondant selon les formats XML (destiné à l'EIAH) et MOF (destiné à la couche d'Intégration). Ainsi, la Figure 5-5 illustre, en langage UML, le modèle correspondant à l'expérience illustrée par la Figure 5-4 proposée plus haut : elle comprend une instance d'un objet *Oscilloscope* ainsi qu'une instance d'un objet *SignalGenerator*, mais également une association *SigGenToScope* qui exprime la relation entre le générateur de fonctions et l'oscilloscope.

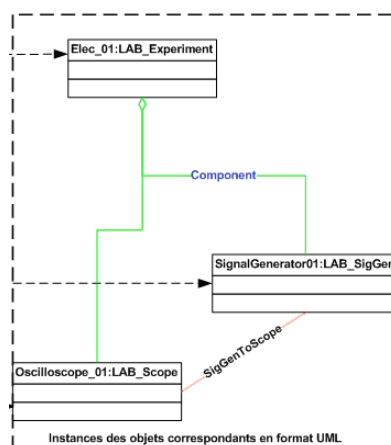


Figure 5-5. Génération du modèle d'une expérience en électronique.

Une fois le modèle élaboré, le concepteur peut passer à la phase de déploiement afin de tester son expérience avant de la soumettre aux utilisateurs finaux (apprenants et tuteurs). Le succès

de cette phase est tributaire de la disponibilité des dispositifs choisis pour l'expérience. Par exemple, le concepteur peut être obligé d'attendre une période de temps nécessaire à la libération des ressources requises pour le déploiement de son expérience avant de la tester réellement. Dans l'exemple précédent, au moins un générateur de fonctions et un oscilloscope doivent être disponibles pour pouvoir déployer l'expérience.

5.2.3.2. L'interface « Cockpit »

Pour mettre en œuvre l'IHM responsable des phases d'exploitation, de sauvegarde et de restauration, nous nous sommes inspirés de la métaphore du « Cockpit » qui repose sur l'analogie entre la mission menée par un étudiant enrôlé dans une activité de travaux pratiques en ligne en face de son ordinateur, et celle d'un pilote dans l'habitacle de son véhicule. Les points communs entre ces deux situations sont la complexité des moyens de contrôle et de supervision d'une part, et la densité de l'IHM d'autre part qui doit être capable d'exposer en une seule interface tous ces moyens d'instrumentation (Gillet et al., 2003).

La Figure 5-6 illustre la proposition de D. Gillet pour une interface « Cockpit ». Cette structure a été établie suite à une étude comparative menée auprès d'étudiants conduisant des TP (en présence) dans trois domaines : l'automatique, la biomécanique et la mécanique des fluides. Elle a permis de déterminer les caractéristiques communes des tâches typiques d'expérimentation effectuées par des étudiants inscrits en troisième et quatrième années du cursus d'ingénieur à l'EPFL³¹ (Ecole Polytechnique Fédérale de Lausanne). Le résultat est une interface divisée en trois principaux espaces, en plus d'une barre de navigation et d'une zone réservée aux informations diverses (voir Figure 5-6) :

- *La console d'expérimentation* constitue la partie interactive de l'interface à travers laquelle l'apprenant réalise l'expérience via les outils de commande et de supervision.
- *La console boîte à outils* contient des outils nécessaires à l'apprenant pour la conduite de son expérience. Dans l'exemple de la Figure 5-6, une interface dédiée à l'apprentissage de la mécatronique fournit des outils de calcul, de génération de graphiques et d'analyses mathématiques.
- *Le journal du laboratoire* est un environnement de rédaction collaboratif de notes et de comptes rendus, et permet ainsi l'intégration et le partage de nouvelles connaissances.

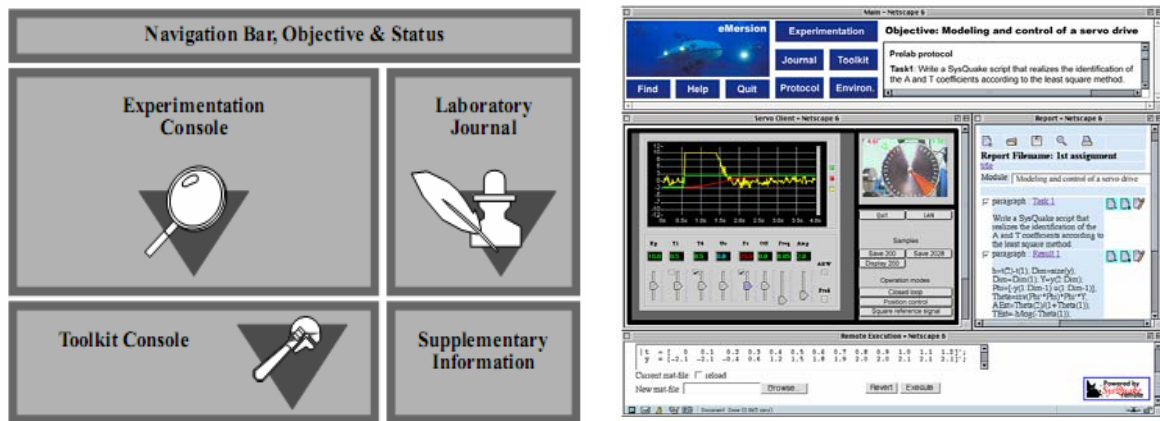


Figure 5-6. Structure générale d'une interface « Cockpit », et son application au domaine de la mécatronique (Gillet et al., 2003).

L'interface « Cockpit » considère en partie le principe de l'*Awareness* de l'espace de travail que nous avons identifié comme prérequis à un apprentissage collaboratif de qualité (voir le chapitre 2), mais elle n'est pas munie de certaines fonctions primordiales à l'efficacité pédagogique d'une activité de télé-TP :

- Aucun espace de travail n'est prévu pour accueillir des outils de communication synchrones ou asynchrones, ni pour favoriser le travail collaboratif ; les apprenants sont donc livrés à eux-mêmes durant l'activité d'apprentissage et ne peuvent bénéficier de l'aide des tuteurs et/ou des autres apprenants.
- Les apprenants disposant d'une interface « Cockpit » ne peuvent accéder à une vision commune de l'expérience, ni visualiser en temps réel les interactions de chaque membre du groupe avec l'expérience ; l'*Awareness* de l'artefact n'est donc pas prise en compte.
- De la même manière, aucun outil n'est offert aux utilisateurs pour visualiser, a posteriori, les travaux réalisés par les apprenants durant une session de télé-TP ; l'absence de support à l'*Awareness* asynchrone restreint les fonctionnalités d'apprentissage d'une part, et limite les capacités de tutorat personnalisé d'autre part.

Afin de combler ces manques, nous proposons certaines recommandations organisationnelles et fonctionnelles aboutissant à une proposition d'interface graphique permettant d'atteindre l'efficacité pédagogique.

³¹ <http://www.epfl.ch/index.fr.html>

5.2.3.3. L'interface d'exploitation, de sauvegarde et de restauration

L'objectif général de nos travaux étant de généraliser et simplifier, avec la plus forte qualité de service, l'accès aux télé-TPs pour les différents acteurs humains, l'interface de contrôle pédagogique doit faire partie intégrante de l'EIAH usuel des utilisateurs. Aujourd'hui, les Environnements Numériques de Travail (ENT) déployés aussi bien dans l'enseignement supérieur que dans les collèges et lycées sont pour la plupart accessibles à travers un réseau, et plus spécifiquement via un navigateur Internet. Même si la structure du « Cockpit » ne se limite pas aux environnements informatiques fondés sur les technologies web, nous préconisons de développer les IHMs de télé-opération à partir de ces technologies. Les IHMs de télé-opération deviennent alors un composant natif des EIAHs existants, puisqu'ils assurent l'unification de toutes les interfaces.

L'intégration de l'IHM de télé-opération dans un environnement d'apprentissage existant apporte de nombreux atouts et services :

- Le télé-TP vient étendre les activités pédagogiques préparées par les enseignants dans un cursus en ligne traditionnel, donc les prérequis à la réalisation d'une expérience donnée peuvent facilement être consultés par les apprenants.
- Les outils de communication synchrones et asynchrones (principalement la messagerie instantanée et les fils de discussion), ainsi que les systèmes de production collaborative (comme le *wiki*), sont nativement proposés par les EIAHs et peuvent être réutilisés pour mettre en œuvre des activités de tutorat et de collaboration.
- Le mécanisme d'authentification d'un utilisateur sur une session de télé-TP, ainsi que la spécification de son rôle, est délégué à la plateforme hébergeant l'IHM de télé-opération.
- L'objet pédagogique interactif « Expérience » peut être joué par les moteurs d'exécution de scénarios pédagogiques nativement implantés dans les EIAHs existants.

De plus, une interface pédagogique dédiée à l'expérimentation en ligne doit être munie des capacités fonctionnelles suivantes :

1. Elle doit fournir une vue unifiée et homogène de tous les composants d'une expérience, indépendamment du niveau de complexité de celle-ci.
2. Elle doit être constituée d'une unique page web, selon le principe du « Cockpit », afin de diminuer la charge cognitive de l'apprenant.

3. Les fonctionnalités proposées à l'utilisateur par l'IHM de télé-opération doivent varier en fonction du rôle de ce dernier au sein de l'EIAH hôte. En particulier, les enseignants et tuteurs doivent disposer d'une interface de supervision globale de l'expérience lors de la session de télé-TP.
4. Elle doit faciliter la classification des sessions en cours selon divers critères (expérience, équipement, type de session, chronologie, etc.) afin de supporter le passage à l'échelle et un grand nombre d'utilisateurs.

Enfin, pour implémenter le principe de l'*Awareness* de l'artefact, l'IHM de télé-opération doit être pourvue de mécanismes permettant de savoir à quel moment « qui fait quoi » et « qui a fait quoi ». Ces fonctionnalités impliquent la mise en œuvre d'outils de partage d'une session entre plusieurs utilisateurs (en lecture seule pour visualiser les actions réalisées par les acteurs impliqués dans la session, ou en écriture afin de rendre possible le travail collaboratif), et d'un système de journalisation, ou Système à Base de Traces (SBT), pour enregistrer les actions opérées par les apprenants afin de rejouer les différentes sessions à n'importe quel moment selon les principes de l'*Awareness* asynchrone. Aussi, en complément du partage d'une session en mode écriture qui offre un espace de travail collaboratif, l'IHM doit donner la liste des utilisateurs ainsi que leur statut (présent, absent, occupé, etc.) pour faciliter la prise de conscience mutuelle.

Dans l'objectif de concilier la complexité de la télé-manipulation d'un nombre important de dispositifs de laboratoire et la nécessité d'assurer la conscience mutuelle au sein du groupe d'apprentissage, l'architecture générale de notre IHM illustrée par la Figure 5-7 s'articule autour d'une fenêtre principale d'expérimentation composée d'outils de télécommande et de télé-supervision de l'expérience en ligne, assortie d'un minimum d'outils de communication et de collaboration. Les cadres latéraux qui entourent la fenêtre principale permettent d'une part d'explorer et de consulter l'état des composants d'une expérience, et d'autre part de visualiser les enseignants, tuteurs et apprenants participant à l'expérience et d'exploiter des outils assurant une collaboration efficace dans ce contexte.

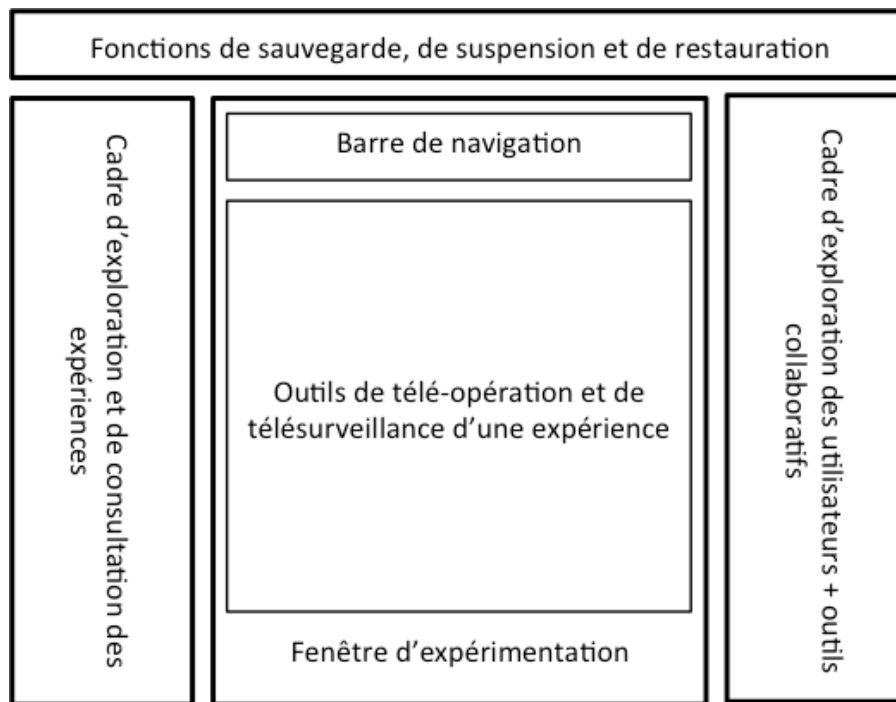


Figure 5-7. Architecture d'IHM pour l'exploitation des télé-TPs.

Le cadre supérieur offre des contrôles qui implémentent les fonctions de suspension/reprise, sauvegarde et restauration de l'expérience en cours. Selon ses privilèges, un utilisateur est en mesure de suspendre l'accès à l'expérience puis de le rétablir alors que cette dernière est opérationnelle, de sauvegarder l'expérience dans son état actuel et de la restaurer dans l'état enregistré.

Afin d'alléger l'IHM et de minimiser la charge cognitive induite par l'affichage d'un grand nombre de boutons, nous avons adopté des menus contextuels spécifiques à chaque élément ou classe d'éléments. Sur la Figure 5-7, le cadre latéral gauche de l'IHM permet d'explorer tous les éléments d'une ou de plusieurs expériences via un contrôle d'affichage arborescent ; ce type de contrôle permet une visualisation réduite ou exhaustive selon le besoin. Ce cadre offre également un accès à un dispositif de l'expérience (en activant l'option adéquate du menu contextuel de cet appareil) afin d'effectuer les (télé)-opérations requises pour achever l'activité de télé-TP. Lorsqu'un utilisateur active cette dernière fonction, l'IHM du dispositif correspondant s'affiche dans la fenêtre d'expérimentation et devient accessible via un onglet de la barre de navigation qui permet de facilement manipuler les différents dispositifs de l'expérience. Ainsi, au lieu d'être submergé par un grand nombre de fenêtres ou consoles,

l'apprenant dispose d'un moyen efficace pour identifier les différents dispositifs qu'il manipule en naviguant parmi les différents onglets qu'il a ouverts.

D'autre part, les enseignants concepteurs et tuteurs disposent de fonctions de contrôle et de supervision plus avancées, cachées aux apprenants pour des raisons pédagogiques et de sécurité. Les menus contextuels affichés à cette catégorie d'utilisateurs permettent par exemple d'effectuer des tâches de réinitialisation (cette fonctionnalité devient utile lorsque l'expérience atteint une situation d'incohérence ou lorsqu'elle doit être mise à disposition d'un nouveau groupe d'apprenants), d'exécution de commandes en lot ou de consultation de l'état global de l'expérience (afin d'identifier rapidement les situations de blocage auxquelles font face les apprenants).

Le cadre latéral droit de l'IHM expose un explorateur d'utilisateurs afin d'identifier les apprenants, enseignants et tuteurs associés à l'expérience, et de connaître leurs états de connexion et de disponibilité. Le menu contextuel associé à l'icône d'un utilisateur permet à la fois d'ouvrir des sessions de travail collaboratives, mais également de consulter, en temps réel ou en différé, l'historique des opérations effectuées par l'utilisateur sélectionné sur un dispositif donné. Cette dernière fonctionnalité relève de la gestion des traces d'activités réalisées par les utilisateurs sur les laboratoires distants et qui est sous la responsabilité de l'agent correspondant externe à l'IHM de télé-opération et présenté dans la section 5.2.2.

Si une implémentation de ce prototype pour l'enseignement du domaine de l'informatique est présentée en détail dans la partie III de ce manuscrit, la Figure 5-8 illustre une IHM pour une expérience en génie électrique fondée sur nos propositions et qui correspond à l'expérience décrite par la Figure 5-4. Ainsi, les différents onglets correspondent aux différents dispositifs associés à l'expérience afin d'être en mesure de les télé-opérer, alors que les cadres latéraux offrent l'opportunité de consulter l'état de ces dispositifs ainsi que les différents utilisateurs participant à l'expérience.

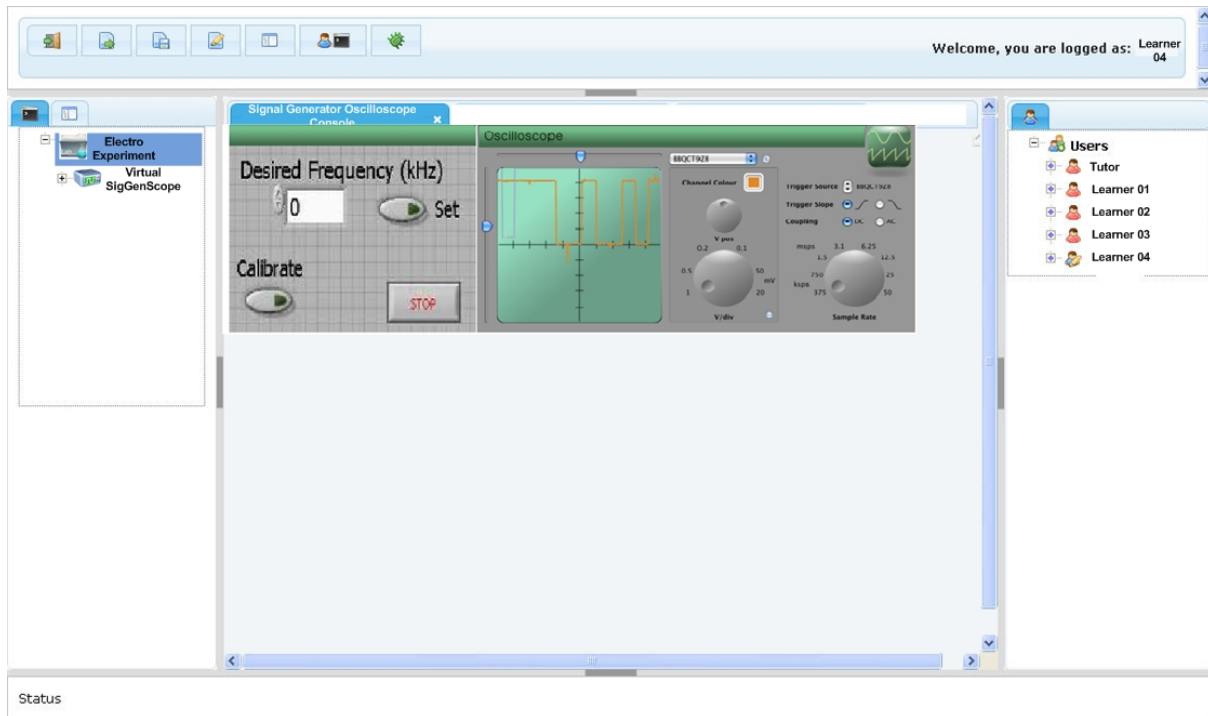


Figure 5-8. Prototype d'IHM pour une expérience en génie électrique.

Nous avons présenté dans cette section les règles générales qui constituent les fondements de la conception de nos prototypes génériques d'IHMs dédiés à la conception, à l'exploitation et au suivi d'expériences en ligne. Nous avons essayé de concilier la richesse des fonctionnalités associée à la complexité des dispositifs manipulés et la malléabilité de l'interface pour éviter la surcharge cognitive tout en assurant l'efficacité pédagogique des activités. Dans la prochaine section nous présentons l'architecture de la couche d'Expérimentation qui renferme les dispositifs concrets, effectivement manipulés dans le cadre d'expériences en ligne.

5.3. La couche d'Expérimentation

La couche d'Expérimentation renferme les différents dispositifs (physiques, virtuels ou hybrides) appartenant à un ou plusieurs laboratoires centralisés ou distribués sur plusieurs sites. Elle offre, pour chacun des dispositifs, une interface d'instrumentation normalisée et unificatrice qui étend les interfaces existantes, et expose cette interface à la couche supérieure via un protocole de communication standard.

Notre contribution consiste à doter chaque dispositif de laboratoire d'un Pilote (voir Figure 5-1) qui rend possible l'instrumentation et la surveillance des dispositifs par la couche

supérieure. Un Pilote est un programme s'exécutant au sein du système d'exploitation du dispositif dont il est responsable lorsque ce dernier est muni d'un système embarqué, sinon il est exécuté dans le système de l'ordinateur chargé de médiatiser le dispositif. Comme les pilotes de périphériques d'un ordinateur, le Pilote d'un dispositif dans notre contexte a la capacité de communiquer avec le dispositif qu'il gère en utilisant les commandes et les structures de données de bas niveau relatives à ce dernier ; il possède également la connaissance du modèle associé au dispositif, élaboré dans un objectif de description et d'instrumentation. Ainsi, les données et commandes de haut niveau reçues de la couche d'Intégration pour les exécuter sur le dispositif sont traduites en routines et structures compréhensibles par ce dernier. Inversement, les messages de réponses et d'événements générés par le dispositif sont traduits en structures de haut niveau compréhensibles par la couche supérieure. Ces échanges sont illustrés par la Figure 5-9.

Les intérêts de la spécialisation de chaque Pilote à chaque dispositif géré sont multiples, parmi lesquels :

- Permettre aux développeurs de se focaliser sur le développement des Pilotes qui requièrent des compétences techniques spécifiques aux dispositifs gérés.
- Constituer une base de données de Pilotes pour chaque équipement ou logiciel de laboratoire afin qu'ils soient réutilisables dans différents contextes.
- Déplacer la gestion des expériences complexes et distribuées à la couche d'Intégration qui supervise l'ensemble des laboratoires (distribués) dont elle est responsable.

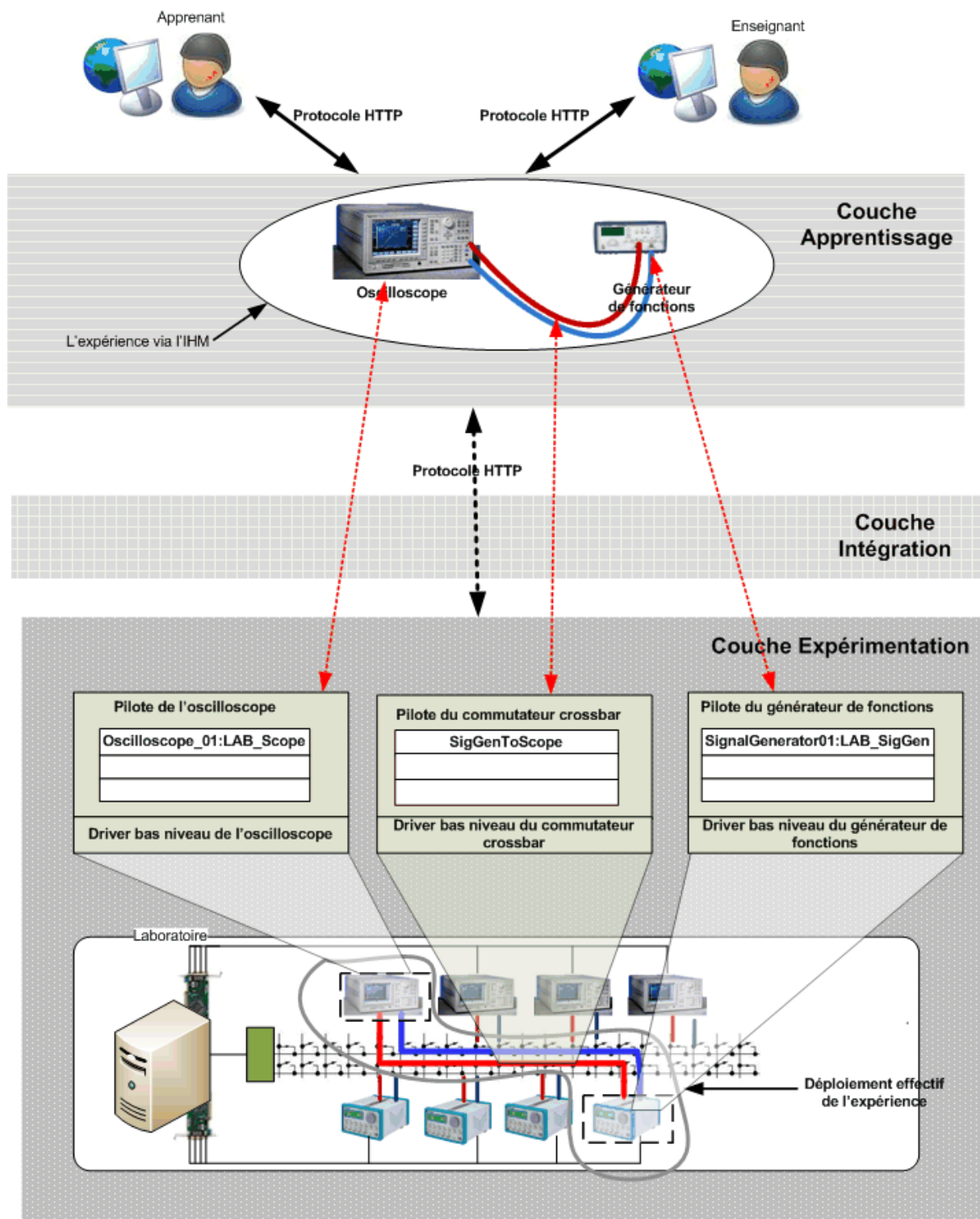


Figure 5-9. Coopération entre l'interface utilisateur et les dispositifs de la couche d'Expérimentation dans le domaine du génie électrique.

Dans la section suivante nous présentons l'architecture générale d'un Pilote de dispositif, puis donnons la description du protocole de communication et de l'interface d'instrumentation par lesquels il est caractérisé.

5.3.1. Architecture des Pilotes de dispositifs

Comme nous l'avons mentionné plus haut, un Pilote implémente le modèle du dispositif qu'il instrumente en associant les propriétés et les opérations exécutables sur ce dernier aux attributs et méthodes de la classe correspondante. L'interaction entre le Pilote et la couche d'Intégration est assurée par l'échange de structures de données de haut niveau, soit par des requêtes d'écriture et/ou de consultation de ses attributs, soit par l'invocation d'une de ses méthodes. Le Pilote traduit alors ces requêtes en opérations de bas niveau compréhensibles et offertes par le dispositif. La Figure 5-10 illustre l'architecture générale d'un Pilote pour les cas les plus fréquents :

- Le Pilote interagit directement avec le dispositif via l'API standard des pilotes des périphériques de communication traditionnels (GPIB, RS232, USB ou Ethernet, Wifi, etc.).
- Le Pilote interagit indirectement avec le dispositif via l'API d'un pilote de haut niveau fourni par un éditeur tiers comme les périphériques virtuels de LabView (NI, 2011) ou MathWorks (MathWorks, 2011).
- Le Pilote est embarqué dans le dispositif géré lorsque ce dernier exécute un système d'exploitation, comme dans le cas des systèmes informatiques classiques (PC, serveurs, machines virtuelles), des équipements de réseaux et télécommunication, des sondes et capteurs sans fil, et d'une manière générale tout dispositif ayant un système informatique embarqué.
- Le Pilote utilise l'API d'instrumentation fourni avec le dispositif, comme dans le cas des robots.

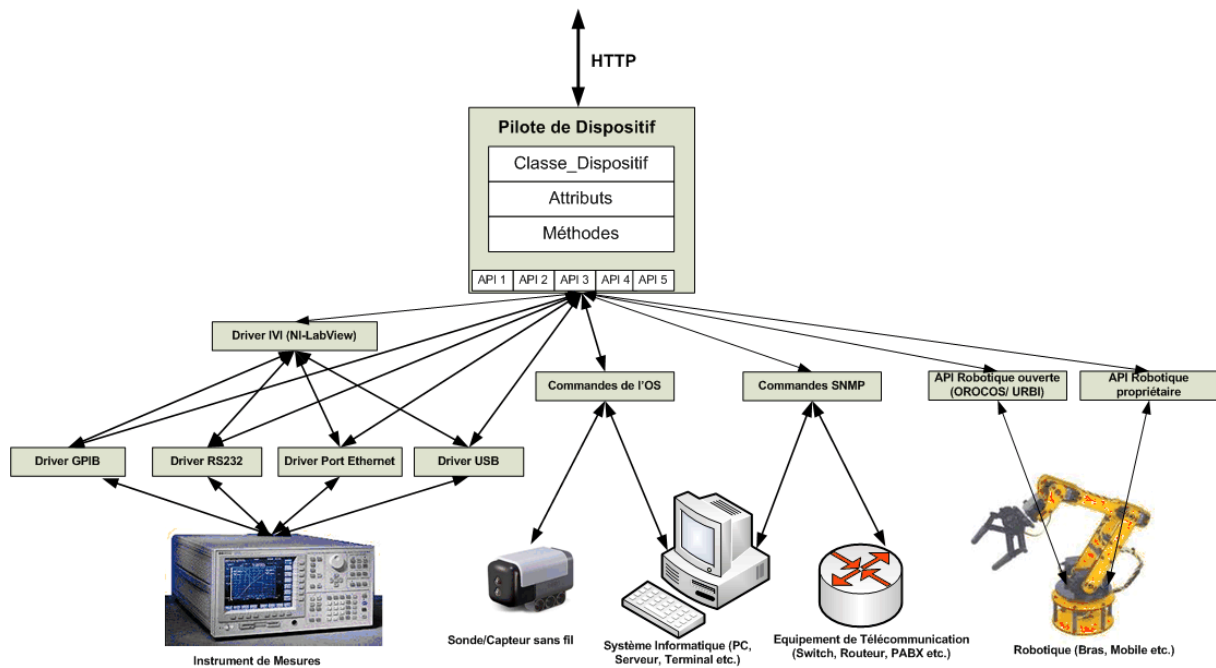


Figure 5-10. Architecture des Pilotes de dispositifs des laboratoires.

5.3.2. Le protocole de communication avec la couche supérieure

Dans la mesure où les environnements d'apprentissage en ligne sont des systèmes fortement distribués, les interactions entre la couche d'Intégration et les composants de la couche d'Expérimentation sont assurées par le protocole HTTP. Ce protocole de transport, utilisé dans le réseau Internet, est le seul à garantir la traversée de réseaux hétérogènes et de pare-feux logiciels ; dans le chapitre suivant, nous décrirons les informations encapsulées dans ce protocole.

Lorsque la couche d'Intégration initie une communication avec un Pilote de dispositif de la couche d'Expérimentation, les informations véhiculées dans la requête HTTP correspondent aux commandes d'instructions formulées par l'utilisateur, et sont représentées dans un format standard. Le rôle du Pilote du dispositif consiste alors à traduire la requête standard en une requête spécifique compréhensible par l'équipement géré. Dans l'autre sens, il traduit les réponses de ce dernier dans le format commun.

5.3.3. Une interface standardisée et unificatrice

La Figure 5-10 ci-dessus donne un aperçu de l'hétérogénéité des moyens d'instrumentation des différents dispositifs de laboratoire. Les seules normes existantes que sont IVI

(*Interchangeable Virtual Instrument*) et SCPI (*Standard Commands for Programmable Instruments*) ne couvrent pour le moment que douze classes d'instruments électroniques. Pour les autres types de dispositifs, l'outil prédominant pour le pilotage est sans doute LabView qui possède sa propre interface propriétaire. Les projets OROCOS (Open Robot Control Software) ou URBI sont des bibliothèques d'API ouvertes dédiées au pilotage de robots, mais d'une manière générale les constructeurs créent leurs propres spécifications souvent fermées et propriétaires.

Notre approche consiste à fédérer les protocoles et interfaces existants, plutôt que de tenter de proposer un nouveau standard de télé-instrumentation. Nous proposons d'exploiter les interfaces existantes à travers une interface intermédiaire normalisée qui expose un ensemble standard de méthodes et de structures de données pour implémenter les routines spécifiques de gestion dédiées à chaque composant télé-instrumenté. La spécification de cette interface sera explicitée dans le prochain chapitre, dans la mesure où elle est utilisée au niveau de la couche d'Intégration.

5.3.4. La problématique du déploiement automatique des expériences

Nous reprenons ici la problématique du déploiement automatique des expériences plus en détails après l'avoir évoqué dans la section 5.2.1 consacrée au cycle de vie de l'objet pédagogique « Expérience ». Rappelons que cette problématique est divisée en deux parties : l'identification et la réservation des ressources existantes correspondant aux besoins d'une expérience (en fonction de son modèle), et l'initialisation de ces ressources pour construire concrètement l'expérience.

L'identification et la réservation des ressources ont été étudiées dans le projet NetBed (NetBed, 2010). Cette étude a montré que ces processus correspondent à un problème mathématique NP-Complet (Rici et al., 2003) dont la résolution d'une façon générique (couvrant toutes les classes de dispositifs) dépasse le cadre de notre travail. Dans notre implémentation, présentée dans la partie III, nous avons mis en œuvre une politique de réservation basique où nous comptons sur la coopération des équipes pédagogique et technique pour assurer l'équité de l'accès aux ressources.

Quant à l'initialisation des ressources allouées à une expérience selon le modèle exprimé par le concepteur, elle dépend des moyens et technologies des dispositifs de laboratoires. Si

l'initialisation des ressources est envisageable pour un grand nombre de dispositifs existants (comme nous l'avons mentionné, de plus en plus de dispositifs embarquent un système informatique permettant leur télé-opération), la difficulté réside dans la mise en œuvre des connexions entre ces divers équipements dans le cadre d'une expérience complexe. La résolution de cette problématique constitue la condition indispensable à l'exploitation efficace et optimale d'infrastructures de laboratoires distribués et mutualisés par des consortiums d'établissements de formation.

La Figure 5-11 illustre cette problématique dans le domaine du génie électrique, où un laboratoire est constitué de plusieurs instruments (oscilloscopes, générateurs de fonctions, ampèremètre, voltmètres, etc.) dont les accès sont assurés par un ou plusieurs serveurs de médiatisation. Une expérience consiste généralement à brancher la sortie d'un instrument vers l'entrée d'un autre instrument, ou effectuer des combinaisons arbitraires de ce type de branchements. Alors pour réaliser ces interconnexions à distance, des dispositifs technologiques spécifiques appelés « Matrices Programmables de Points Croisés³² » ont été conçus (voir Figure 5-11). Dans notre approche, et en reprenant l'expérience illustrée par la Figure 5-2, le scénario de déploiement consiste à effectuer les deux opérations suivantes :

- Programmer le commutateur pour diriger la sortie du générateur de fonctions vers l'entrée de l'oscilloscope.
- Initialiser les composants de l'expérience avec les paramètres de démarrage choisis par le concepteur.

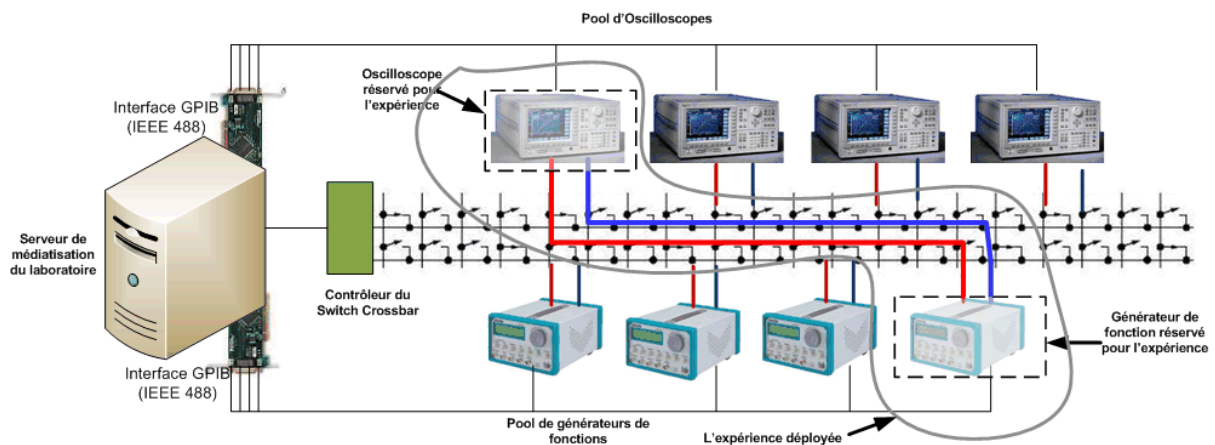


Figure 5-11. Déploiement d'une expérience dans un laboratoire.

³² Traduction libre des termes: Crossbar Switch/Matrix, Cross-Points Switch/Matrix

A l'inverse, l'arrêt de l'expérience se traduit par la libération des ressources allouées. Il s'agit donc de :

- Programmer le commutateur pour libérer les connexions entre le générateur de fonctions et l'oscilloscope.
- Initialiser les composants de l'expérience avec les valeurs par défaut.

5.4. Synthèse

Nous avons présenté dans ce chapitre les couches supérieure et inférieure de notre architecture globale, en détaillant les caractéristiques fonctionnelles et organisationnelles de chacune d'entre elles. Nos propositions apportent les atouts suivants à la couche d'Apprentissage :

- Elle ne connaît pas, et n'a pas à se soucier, des détails techniques d'instrumentation des dispositifs de laboratoires distants.
- Elle dispose d'une vue générique, voir abstraite, des environnements de laboratoires afin de pouvoir les exploiter d'une façon transparente.
- Elle ignore la localisation des laboratoires distants dont le caractère distribué est totalement masqué.
- Elle offre des IHMs de conception, d'apprentissage et de suivi des expériences en ligne malléables et personnalisables, conciliant légèreté et richesse des fonctionnalités.

La couche d'Expérimentation offre quant à elle une interface d'exploitation unique et homogène pour n'importe quel objet d'expérimentation. Ainsi, elle homogénéise et simplifie les moyens d'instrumentation tout en restant indépendante des environnements d'apprentissage en ligne.

Dans les prochains chapitres, nous présentons la couche d'Intégration qui constitue le cœur de toute l'architecture car elle implémente toute l'intelligence et la complexité de la gestion des laboratoires, des ressources de laboratoires et des expériences sur ces laboratoires.

Chapitre 6. La couche d'Intégration

6.1.	<u>Les services offerts à la couche d'Apprentissage</u>	136
6.1.1.	<u>Le service de gestion des laboratoires distants</u>	136
6.1.2.	<u>Les services de gestion des expériences en ligne</u>	137
6.1.3.	<u>Le service d'authentification et d'autorisation</u>	141
6.1.4.	<u>Le service de gestion des traces d'activités</u>	142
6.2.	<u>Les composants d'intégration des laboratoires en ligne</u>	143
6.2.1.	<u>Les fondements de notre approche</u>	144
6.2.2.	<u>Le serveur WBEM primaire</u>	148
6.2.3.	<u>Les serveurs WBEM secondaires</u>	152
6.2.4.	<u>Les Pilotes des dispositifs de laboratoires</u>	153
6.2.5.	<u>Le protocole d'échange d'informations</u>	153
6.3.	<u>Synthèse</u>	154

L'architecture de la couche d'Intégration est inspirée des intergiciels (ou *middleware*) de gestion d'infrastructures de grilles de calcul comme GLITE³³, dont les objectifs de partage et de mutualisation de ressources de calcul pour des programmes intensifs présentent de nombreux points communs avec nos travaux. Notre architecture constitue une variante plus légère adaptée à l'enseignement en ligne, qui permet un passage à l'échelle rapide, un équilibrage de charge entre les ressources, et un service minimum en cas de panne ou de catastrophe.

Cette couche, dont les détails sont illustrés par la Figure 6-1, est au cœur de notre architecture car elle assure la communication transparente entre les couches d'Apprentissage et d'Expérimentation présentées dans le chapitre précédent. Son niveau supérieur est constitué d'un ensemble de services exposés aux EIAHs et dédiés à la gestion et à l'exploitation des laboratoires distants, alors que son niveau inférieur, responsable de l'instrumentation de ces

³³ <http://www.glite.org>

services, possède de nombreuses caractéristiques : (i) il renferme, de façon uniforme, la connaissance de tous les laboratoires distants, (ii) il communique avec les laboratoires afin d'ordonner des instructions ou obtenir différentes informations de supervision, (iii) il reçoit des notifications issues des ressources des laboratoires, et (iv) il renferme les traces d'activités générées par les actions effectuées par les utilisateurs au sein de l'IHM de télé-opération.

Dans le niveau inférieur, nous distinguons, selon les principes des organisations virtuelles, les concepts de site central et de sites locaux : au sein d'un ensemble de sites appartenant à des établissements partenaires, un site central est chargé de la coordination et de la centralisation des informations liées à la gestion des ressources et distribuées sur les différents sites locaux. Ainsi, pour maintenir un état cohérent de l'utilisation des ressources des laboratoires distants, des expériences qui y sont déployées et des traces d'activités réalisées par les acteurs humains, le niveau inférieur repose sur une base de connaissances centralisée pour stocker l'ensemble de ces informations au niveau du site central. Cette base centrale est soutenue par d'autres bases de connaissances locales situées au niveau des autres sites et possédant les mêmes caractéristiques, mais à un niveau local.

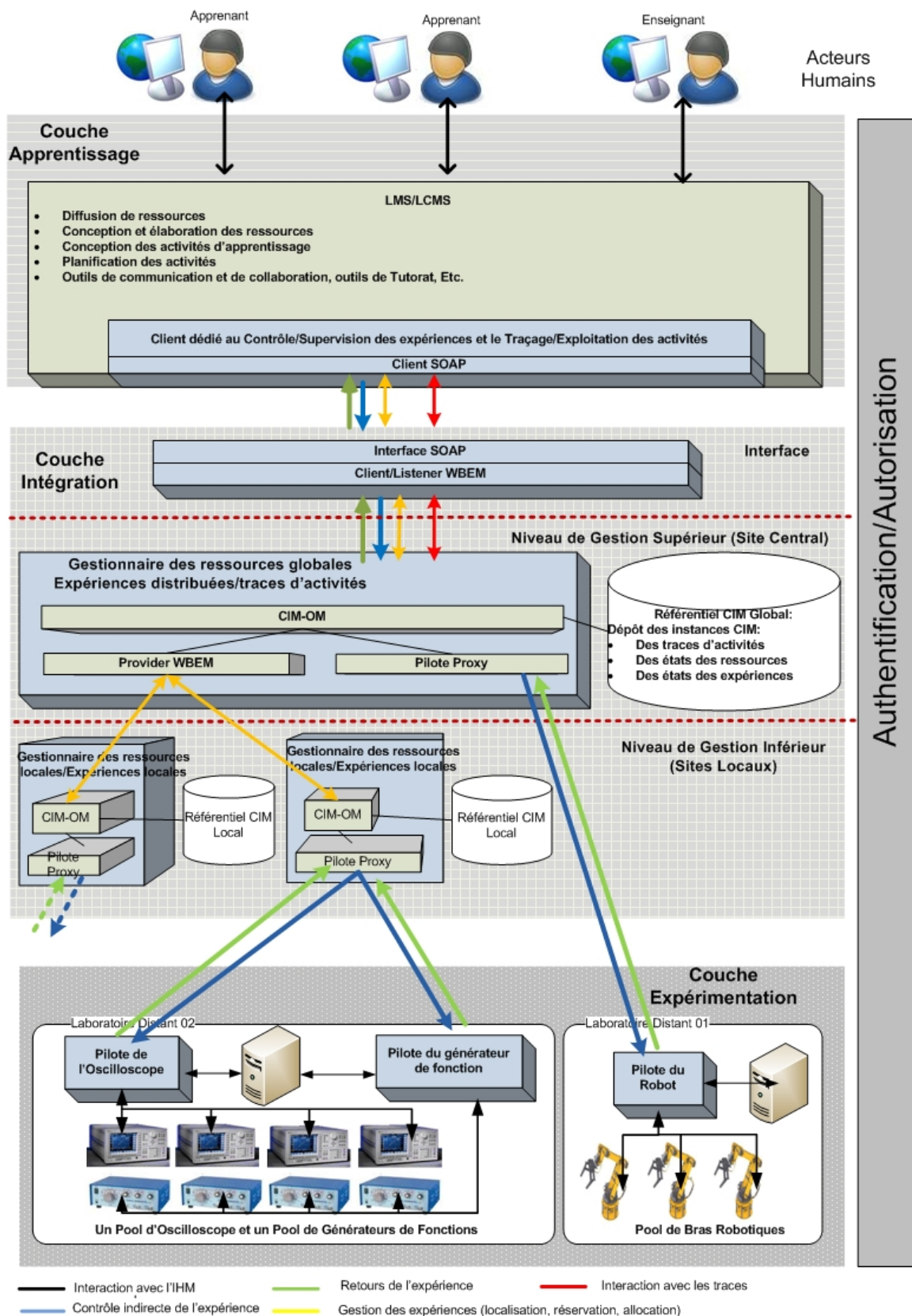


Figure 6-1. Détails de la couche d'Intégration.

6.1. Les services offerts à la couche d'Apprentissage

Le niveau supérieur de la couche d'Intégration correspond à la partie visible vis-à-vis de la couche d'Apprentissage. Elle permet l'exploitation des expériences en laboratoires à distance par les EIAHs à travers un ensemble de services liés à la gestion des laboratoires distants et des expériences, à la gestion des mécanismes d'authentification et d'autorisation, et au suivi des activités. Ce niveau est fondé sur les principes des Architectures Orientées Services (*Service Oriented Architecture* - SOA) afin de satisfaire le caractère distribué de notre architecture et masquer les différentes fonctionnalités implémentées par les couches d'Intégration et d'Expérimentation. Dans ce qui suit nous présentons les quatre services que nous proposons afin de couvrir l'ensemble des fonctionnalités requises aux activités de télé-TP : (1) le service dédié à la gestion des laboratoires distants, (2) le service dédié à la gestion des expériences en ligne, (3) le service dédié à la gestion de l'authentification et de l'autorisation, et enfin (4) le service dédié à la gestion des traces d'activités.

6.1.1. Le service de gestion des laboratoires distants

Le service de gestion des laboratoires vise deux principaux objectifs : permettre la fédération et la mutualisation de laboratoires en offrant la possibilité d'ajouter, de modifier ou de supprimer des laboratoires d'une façon homogène et transparente, et faciliter le développement d'applications d'administration des laboratoires distants. Les méthodes associées à ce service sont présentées dans le Tableau 6-1.

Tableau 6-1. Spécification des méthodes du service de gestion des laboratoires.

Méthode	Paramètres d'entrée	Fonctions
RegisterLab	Array[] labProperties Array[] resourcesProperties	Création d'un laboratoire dans la base de connaissances et initialisation de ses attributs (voir Tableau 7-1). Création des ressources dans la base de connaissances et initialisation de leurs attributs (voir Tableau 7-2). Retourne <i>True</i> en cas de succès, <i>False</i> sinon.
ModifyLabStatus	Array[] labProperties Array[] resourcesProperties	Mise à jour dans la base de connaissances du statut d'un laboratoire ou des ressources dont il dispose. Retourne <i>True</i> en cas de succès, <i>False</i> sinon.
DeleteLab	String labId	Suppression, dans la base de connaissances, d'un laboratoire et des ressources qu'il renferme.

		Retourne <i>True</i> en cas de succès, <i>False</i> sinon.
GetLabsList	Void	Retourne la liste des identifiants des laboratoires enregistrés dans la base de connaissances, ainsi que la localisation des entités responsables de leur gestion.
GetLabStatus	String labId	Retourne l'état d'allocation des ressources d'un laboratoire sous la forme d'un tableau contenant, pour chaque type de ressources, le nombre d'instances occupées et le nombre d'instances disponibles.
AllocateLabResource	String labId String resourceType Integer nbInstances Array[] resourceIds (option)	Modifie l'état de(s) la ressource dans la base de connaissances (état « allocated »). Retourne la liste des identifiants des ressources réservées.
FreeLabResource	String labId String resourceType Integer nbInstances Array[] resourceIds (option)	Modifie l'état de(s) la ressource dans la base de connaissances (état « available »). Retourne <i>True</i> en cas de succès, ou <i>False</i> sinon.

Ces méthodes offrent l'opportunité à une institution de formation d'enregistrer un laboratoire dans la base de connaissances en précisant les ressources qu'il propose. Ainsi, les EIAHs perçoivent un seul et unique laboratoire virtuel au lieu d'un ensemble « d'îlots » de laboratoires physiques. Un nombre important d'expériences, ou des expériences plus complexes peuvent ainsi être déployées en s'affranchissant des contraintes matérielles et logicielles de chaque site. En effet, la centralisation de l'enregistrement des laboratoires permet par exemple la mise en œuvre d'expériences nécessitant des ressources non disponibles sur un seul site, ou celles exigeant que les ressources soient réellement distribuées sur le réseau (comme dans le domaine des réseaux et des télécommunications).

6.1.2. Les services de gestion des expériences en ligne

Ces services sont destinés au développement d'IHMs et autres outils supports aux tâches d'expérimentation. Ils sont donc exploités par la couche d'Apprentissage, et plus précisément

par le moteur d'exécution des objets « Expérience », pour intégrer de façon transparente les expériences comme ressources dans des activités d'apprentissage scénarisées.

Deux services mettent en œuvre la gestion des expériences : le service de gestion du cycle de vie des expériences, et le service de télé-opération des expériences. Le service de gestion du cycle de vie, dont les méthodes apparaissent dans le Tableau 6-2, assure le suivi global des expériences à travers les fonctions de création, de démarrage, de suspension, de redémarrage et de suppression d'une expérience. Les fonctions de sauvegarde et de restauration étendent la durée de vie de l'expérience sur plusieurs séances de travail, et en particulier en dehors des horaires habituels. De plus, elles favorisent l'apprentissage par essais et erreurs qui est indispensable aux activités de télé-TPs.

Tableau 6-2. Spécification des méthodes du service de gestion du cycle de vie des expériences.

Méthode	Paramètres d'entrée	Fonctions
CreateExperiment	String expModel	Création d'une instance du modèle d'expérience dans la base de connaissances, spécifié au format MOF par le paramètre expModel. Ce processus opère divers algorithmes afin d'identifier les ressources nécessaires, de rechercher les ressources correspondantes et disponibles dans la base de connaissances, et enfin d'invoquer les services de réservation des ressources. Retourne l'identifiant de la nouvelle instance en cas de succès, ou NULL en cas d'échec.
GetExperimentModel	String expId	Recherche du modèle de l'expérience identifiée par expId. Retourne la liste des instances de classes et de relations qui constituent l'expérience en cas de succès, NULL en cas d'échec.
StartExperiment	String expId	Mise à jour du statut des ressources réservées pour l'expérience au sein de la base de connaissances (état « started »). Retourne <i>True</i> en cas de succès, ou <i>False</i> sinon.
SuspendExperiment	String expId	Mise à jour du statut des ressources de l'expérience au sein de la base de connaissances (état « suspended »). Retourne <i>True</i> en cas de succès, ou <i>False</i> sinon.

ResumeExperiment	String expId	Mise à jour du statut des ressources de l'expérience au sein de la base de connaissances (état « <i>started</i> »). Retourne <i>True</i> en cas de succès, ou <i>False</i> sinon.
DeleteExperiment	String expId	Suppression d'une expérience dans la base de connaissances et mise à jour du statut des ressources qui y sont associées (état « <i>free</i> »). Retourne <i>True</i> en cas de succès, ou <i>False</i> sinon.
SaveExperiment	String expId	Sauvegarde, dans la base de connaissances, d'une copie de l'état de l'expérience à partir des informations renfermées dans cette même base et invoque la méthode SuspendExperiment. Retourne l'identifiant de la sauvegarde qui sera utilisée par la méthode de restauration, ou 0 en cas d'échec.
RestoreExperiment	String expId String expBackupId	Mise à jour du statut de l'expérience (et de ses ressources) conformément aux informations contenues dans le modèle de sauvegarde. Retourne <i>True</i> en cas de succès, ou <i>False</i> sinon.
GetExperimentsList	Void	Retourne la liste de toutes les expériences présentes dans la base de connaissances (qu'elles soient arrêtées ou en cours d'exécution).
GetExperimentStatus	String expId String resourceId (option)	Retourne les propriétés d'une expérience donnée (et de ses ressources) à partir des informations de la base de connaissances en cas de succès, ou un message d'erreur en cas d'échec. Si l'identifiant d'une ressource est fourni, retourne uniquement l'état de la ressource spécifiée.

Les deux dernières méthodes exposées ci-dessus facilitent l'élaboration de tableaux de bord implémentant des mécanismes d'observation. Le suivi de l'évolution d'expériences complexes, mais également de laboratoires, devient aisément disponible à travers des interfaces de visualisation avancées et homogènes qui représentent des entités hétérogènes.

Afin d'agir sur les expériences, le service de télé-opération offre deux opérations basiques spécifiées dans le Tableau 6-3 et qui répondent à la problématique de l'hétérogénéité des moyens de télé-instrumentation au niveau de la couche d'Apprentissage. La méthode *ExecuteCommand* est invoquée par les systèmes de la couche d'Apprentissage pour demander

l'instruction d'une commande sur une expérience donnée ; elle reçoit en retour les résultats engendrés par l'exécution de la commande. Le développement d'outils de télé-instrumentation d'expériences, et en particulier d'IHMs de télé-opération, se trouve ainsi facilité. La méthode *ReceiveMessage*, quant à elle, est dédiée à un composant du niveau inférieur de la couche d'Intégration pour recevoir des notifications d'évènements se produisant sur les expériences. Ces deux méthodes sont mises en œuvre à travers l'interaction avec les Pilotes des dispositifs présentés dans la section 5.3.

Tableau 6-3. Spécification des méthodes du service de télé-opération des expériences.

Méthodes	Paramètres d'entrée	Fonctions
ExecuteCommand	String expId String command Array[] commandArgs	Exécution d'une commande sur une ou plusieurs ressources de l'expérience. Retourne les informations produites par la(les) ressource(s) en cas de succès, ou un message d'erreur sinon.
ReceiveMessage	String expId	Reçoit un événement survenu sur l'expérience sous la forme d'une notification.

Notons que les services de gestion des expériences en ligne mettent en œuvre les fonctions d'un Objet Pédagogique Interactif (OPI) qui ont été définies dans le chapitre 4 (pour rappel, il s'agit des fonctions de Référence, d'Observabilité, d'Inspectabilité et de Scriptabilité). Le Tableau 6-4 établit la correspondance entre ces fonctions et les méthodes spécifiées dans les deux tableaux précédents.

Tableau 6-4. Correspondance entre fonctions d'un OPI « Expérience » et les méthodes spécifiées par les services de gestion des expériences.

Fonction d'un OPI	Définition	Méthodes correspondantes
Référence	Fournir la liste des variables de l'OPI.	GetExperimentModel, GetExperimentList, GetExperimentStatus
Observabilité	Signaler l'occurrence d'évènements sur l'OPI.	ReceiveMessage
Inspectabilité	Délivrer, à la demande, les propriétés de l'OPI.	GetExperimentStatus, GetExperimentsList
Scriptabilité	Modifier l'état de l'OPI.	CreateExperiment, StartExperiment,

		SuspendExperiment, SaveExperiment, ResumeExperiment, RestoreExperiment, DeleteExperiment, ExcuteCommand
--	--	---

6.1.3. Le service d'authentification et d'autorisation

Dans notre contexte d'étude, les services d'authentification et d'autorisation des utilisateurs sont délégués au système d'apprentissage, c'est-à-dire gérés par des mécanismes externes à la couche d'Intégration. Toutefois, cette dernière doit se tenir informée des autorisations attribuées aux utilisateurs en vue d'appliquer sa politique de contrôle d'accès sur les laboratoires distants, et par conséquent sur les expériences en ligne.

Le service que nous introduisons ici est donc invoqué par les environnements d'apprentissage en général, et non par les IHMs de télé-opération en particulier, afin de maintenir à jour la connaissance de la couche d'Intégration :

- La méthode *ManageIdentity* enregistre un utilisateur et un système d'apprentissage dans la base de connaissances, ou met à jour le statut de connexion de cet utilisateur sur ce système. Cette méthode est donc invoquée à chaque connexion et déconnexion d'un utilisateur sur un système d'apprentissage.
- La méthode *AttachUserToExperiment* associe un utilisateur ou un groupe d'utilisateurs à une ou plusieurs expériences, et affecte les droits d'accès des utilisateurs en fonction de leur rôle dans l'environnement d'apprentissage. Un usager possède des privilèges de contrôle s'il est décrit comme un apprenant dans l'EIAH, alors qu'un utilisateur bénéficiera en plus des droits de supervision s'il apparaît comme un personnel enseignant au sein du cursus de formation ; le rôle d'un utilisateur est obtenu à partir des informations disponibles au sein du système d'apprentissage.

Le traitement opéré par la méthode ci-dessus est très important car il détermine la vue dont dispose chaque utilisateur sur l'environnement d'expérimentation. Cette méthode est automatiquement invoquée par un EIAH dès lors qu'une activité de télé-TP est intégrée dans un cursus de formation ; l'EIAH extrait, au préalable, la liste des utilisateurs enregistrés dans le cursus, ainsi que leur rôle.

6.1.4. Le service de gestion des traces d'activités

Nous introduisons ici un service capable d'enregistrer et de retrouver, au sein du référentiel de connaissances, les traces d'activités réalisées par les utilisateurs sur une expérience donnée. Les outils d'expérimentation et d'exploitation des traces invoquent les deux méthodes implantées dans ce service pour mettre en œuvre le principe de suivi pédagogique en temps réel ou différé.

Une requête à la méthode *TrackUserActivity* est transmise par l'Agent de gestion des traces d'activités associé à l'IHM de télé-opération à chaque fois qu'un utilisateur opère une commande. Le traitement associé à cette méthode consiste à enregistrer dans le référentiel de connaissances les informations traduisant les actions des utilisateurs ; ces données sont exposées dans le Tableau 6-5 qui décrit la spécification des méthodes du service de gestion des traces. Notons que les réponses issues des expériences en réaction aux commandes réalisées par les utilisateurs, ainsi que les notifications d'évènements ou d'alertes, sont également enregistrées dans la base de connaissances, mais un autre composant de la couche d'Intégration présenté dans la section 6.2 assure cette tâche.

Tableau 6-5. Spécification des méthodes du service de gestion des traces d'activités.

Méthodes	Paramètres d'entrée	Fonctions
TrackUserActivity	String userId String expId String resourceId String command	Enregistre dans la base de connaissances l'activité (argument command) réalisée par l'utilisateur sur une ressource d'une expérience à partir de l'IHM intégrée dans un EIAH. Retourne <i>True</i> en cas succès, ou <i>False</i> sinon.
ReadUserActivity	String expId String userId (option)	Retourne l'ensemble des activités réalisées sur une expérience (ou seulement celles d'un utilisateur spécifique) et qui sont enregistrées dans la base de connaissances.

La méthode *ReadUserActivity* permet de retrouver les traces d'activités d'un utilisateur ou d'un groupe d'utilisateurs (apprenants ou tuteurs) à partir de l'identifiant d'une expérience, et éventuellement de celui d'un utilisateur. Elle retourne l'ensemble des traces enregistrées dans la base de connaissances qui comprennent les actions des utilisateurs ainsi que les réponses issues des expériences. Cette méthode favorise la reconstitution des activités des apprenants durant une session de télé-TP selon deux modes :

- Le mode synchrone, pour suivre en temps réel l'activité de l'apprenant ou du groupe d'apprenants sur une expérience.
- Le mode asynchrone pour analyser, a posteriori, la séquence d'activités d'un apprenant ou de l'ensemble des apprenants sur une expérience.

Ce service joue donc un rôle très important dans le support de l'apprentissage collaboratif et du tutorat en ligne, à la fois en modes synchrone et asynchrone. Il offre l'opportunité de développer des interfaces qui mettent en œuvre les principes de l'*Awareness* du contexte et des artefacts (voir chapitre 2).

Nous avons décrit dans cette section le niveau supérieur de la couche d'Intégration, dont les fonctionnalités couvrent la gestion des laboratoires, expériences, et traces d'activités. Pour implémenter l'ensemble de ces fonctions, ces services s'appuient sur les composants de gestion du niveau inférieur qui interagissent directement avec la base de connaissances, mais surtout avec les expériences en laboratoires distants.

6.2. Les composants d'intégration des laboratoires en ligne

Le système de gestion central du niveau inférieur de la couche d'Intégration implémente tous les processus métiers qui assurent le traitement des flux de données liés à la gestion des ressources des expériences en laboratoires distants durant toutes les phases de leur cycle de vie. Il reçoit des instructions provenant des services du niveau supérieur, ou des notifications d'évènements survenus sur les laboratoires de la couche d'Expérimentation. Dans le premier cas, il transmet les commandes des utilisateurs aux Pilotes embarqués dans les dispositifs de laboratoires distants (directement ou indirectement à travers les systèmes de gestion locaux) qui finalement réalisent concrètement les instructions sur les expériences, et reçoit, dans l'autre sens, les notifications émises par ces mêmes Pilotes. D'autre part, ce niveau inférieur détient la base de connaissances et garantit la consistance et l'intégrité des données qu'elle renferme ; la description détaillée de ces informations est présentée dans le chapitre suivant.

La principale problématique que doivent traiter les composants de gestion de la couche d'Intégration concerne la communication avec des laboratoires distants hétérogènes dans des objectifs de télé-opération et de surveillance. Des éléments de réponse trouvent leurs origines dans les fondements de la Gestion des infrastructures de réseaux informatiques et de

télécommunication. En effet, les objectifs visés par la Gestion consistent à superviser un parc réparti d'équipements physiques et d'entités logiques hétérogènes, en s'appuyant sur des solutions intégratrices pour détecter les surcharges ou dysfonctionnements de certaines entités et effectuer des opérations de maintenance de façon automatisée. Dans notre contexte, il s'agit de superviser les laboratoires en ligne afin de connaître leurs états, et offrir l'opportunité de réaliser des actions sur les équipements qui les composent. L'application des concepts fondamentaux de la Gestion au domaine des télé-TPs adresse de nombreuses problématiques liées à la diversité des domaines scientifiques concernés par ce type d'activité d'apprentissage.

6.2.1. Les fondements de notre approche

Différentes approches de gestion comme *SNMP*³⁴ et *CMIP*³⁵ existent, mais le standard de gestion *Web Based Enterprise Management* (WBEM) élaboré par le *Distributed Management Task Force* (DMTF) représente l'évolution la plus significative de ces approches. En plus de proposer une approche et un formalisme de modélisation pour la spécification d'un modèle d'informations unificateur (qui constituent les fondements de notre base de connaissances), WBEM introduit une architecture fonctionnelle définissant les entités intervenant dans la Gestion, leurs répartitions, les fonctions qu'elles assurent et leurs rôles respectifs. L'architecture fonctionnelle WBEM est articulée autour de quatre composants clés présentés dans la Figure 6-2 :

- Le *Référentiel* représente la base de données des informations de Gestion, et comprend une représentation abstraite homogène de l'ensemble des éléments gérés dans le parc informatique. Il est utilisé pour implémenter la base de connaissances de notre architecture.
- Le *Gestionnaire WBEM*, ou *Object Manager*, est le composant central de l'architecture. Il assure l'intégrité des données renfermées dans le référentiel et gère les échanges d'informations entre les différents composants de l'architecture.
- Le *Provider WBEM* (distant) et le *Proxy Provider WBEM* sont des entités qui communiquent directement avec le système géré pour le superviser (c'est-à-dire capturer

³⁴ *Simple Network Management Protocol*

³⁵ *common management information protocol*

les valeurs de certaines propriétés comme la charge CPU) et effectuer certaines opérations de Gestion (comme modifier la configuration IP du système géré). Ces opérations sont effectuées suite à une requête formulée par le gestionnaire WBEM, puis le *Provider* lui retourne les résultats des opérations réalisées sur le système géré. Le *Provider* est localisé au sein du gestionnaire WBEM lorsque ce dernier est exécuté dans le système géré, mais il est généralement embarqué dans le système géré qui est réparti sur le réseau. Dans ce dernier cas, le *Provider Proxy* exécuté au niveau du gestionnaire est chargé d'établir la communication avec le *Provider* autonome distant.

- Le *client WBEM* représente les applications de Gestion qui offrent divers services pour superviser les systèmes gérés. Un client WBEM peut envoyer des requêtes au gestionnaire pour consulter ou modifier l'état d'un système géré, mais il peut également souscrire à des événements et recevoir des notifications lorsqu'elles se produisent.

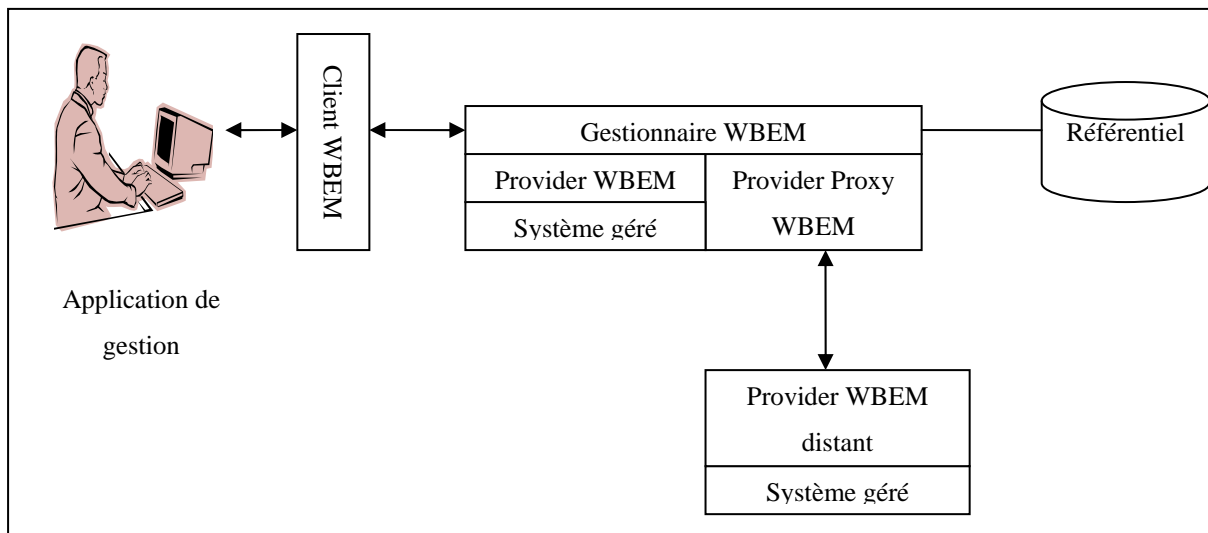


Figure 6-2. Composants clés de l'architecture WBEM.

Les propositions du DMTF sont fortement utilisées dans les domaines de l'informatique, des télécommunications, des réseaux de stockage ou des réseaux électriques (Hobbs, 2004) ; elles sont d'ailleurs implantées nativement dans la plupart des systèmes d'exploitation courants (Windows Management Instrumentation ou WMI pour les systèmes Microsoft, OpenWBEM pour de nombreuses distributions Linux, ou Apple Remote Desktop pour les systèmes Apple). Mais l'approche WBEM ne se limite pas à ces domaines, car elle vise la gestion de tout type d'entités physiques ou virtuelles, pourvu qu'elles soient instrumentées par un système informatique (Hobbs, 2004).

Dans notre contexte, les laboratoires et expériences exploités dans le cadre d'une activité de télé-TP représentent les systèmes gérés. Ainsi, l'approche WBEM facilite d'une part la constitution d'une vue homogène des équipements hétérogènes physiques ou logiques constituant les laboratoires, et d'autre part la mise à disposition de cette vue via le gestionnaire WBEM et les clients associés.

Nous distinguons dans nos travaux les expériences centralisées des expériences distribuées : une expérience centralisée, ou locale, repose sur des dispositifs disponibles dans un laboratoire unique, alors qu'une expérience distribuée, ou délocalisée, exploite des ressources mises à disposition par plusieurs laboratoires appartenant à des organismes de formations différents ayant convenu de mutualiser et de partager leurs ressources ; la Figure 6-3 illustre les concepts d'expériences centralisée et distribuée.

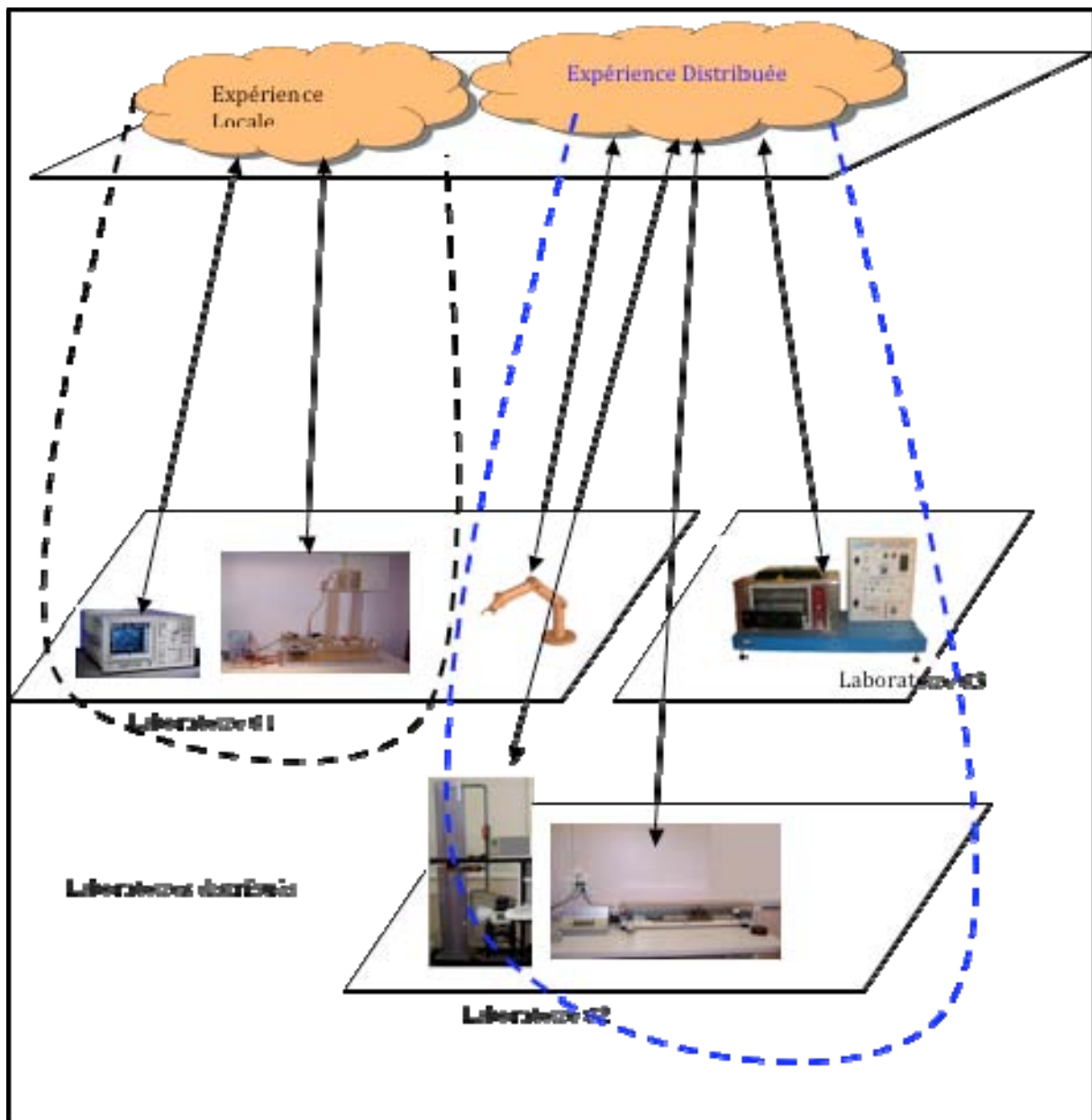


Figure 6-3. Expérience centralisée versus expérience distribuée.

La fédération de différents laboratoires distribués vise d'un côté à satisfaire le maximum d'apprenants et à optimiser l'utilisation des dispositifs disponibles, et d'un autre côté à satisfaire aux besoins d'expériences complexes nécessitant un nombre important de dispositifs, ou des dispositifs trop coûteux pour une seule institution.

Les expériences distribuées engendrent donc parfois une gestion de l'allocation des ressources plus complexe, du fait que le gestionnaire WBEM doit récolter l'état des ressources de tous les sites avant de prendre la décision de réserver les ressources nécessaires à une expérience

donnée. Alors notre approche propose une structuration hiérarchique à deux niveaux pour atteindre l'objectif de mutualisation de ressources partagées par plusieurs institutions :

- Un serveur WBEM primaire est rattaché au laboratoire d'une institution particulière. Son rôle est de piloter et surveiller les ressources de ce laboratoire, mais également de fédérer, d'une manière globale, toutes les ressources de laboratoires partagés par d'autres institutions.
- Des serveurs WBEM secondaires sont chargés de la gestion des ressources hébergées par un laboratoire, et exposent au serveur WBEM primaire les informations de gestion relatives aux ressources dont ils sont responsables. Ce niveau est donc optionnel, mais nécessaire à l'intégration des laboratoires distribués dans une infrastructure unique de laboratoire virtuel.

6.2.2. Le serveur WBEM primaire

Le serveur WBEM primaire est un composant obligatoire dans notre architecture puisqu'il est chargé de constituer un état global de l'utilisation des ressources partagées, et de satisfaire aux demandes de déploiement d'expériences simples ou distribuées selon certains critères et contraintes exprimés dans ces demandes.

6.2.2.1. Le gestionnaire WBEM

Le gestionnaire WBEM est le véritable point d'accès à l'environnement d'expérimentation. Il est exploité par les services du niveau supérieur de la couche d'Intégration qui intègrent un client WBEM, et opérationnalise les traitements associés aux différentes méthodes de ces services :

- Gestion et allocation de ressources pour des expériences centralisées ou distribuées : en fonction de l'état des laboratoires enregistrés dans le référentiel, le gestionnaire identifie les ressources disponibles capables de servir une expérience donnée. Lorsque les dispositifs disponibles ne suffisent pas à déployer une nouvelle expérience, cette dernière reste en attente.
- Déploiement d'expériences simples ou distribuées : après la phase de réservation de ressources, vient la phase de déploiement d'expériences. Elle consiste à initialiser les ressources sur le(s) laboratoire(s) correspondant, en respectant la configuration et les

paramètres exprimés par l'utilisateur lors de la création de l'expérience. A chaque fois qu'une expérience est déployée, le gestionnaire met à jour le référentiel en modifiant les informations des ressources qu'elle utilise.

- Exploitation et supervision d'expériences centralisées ou distribuées : le gestionnaire contrôle et supervise les ressources du (des) laboratoire(s) qu'il gère en enregistrant l'évolution de l'état de chaque expérience dans le référentiel.
- Gestion des traces d'activités des différents acteurs humains : le gestionnaire enregistre (et retrouve) dans le référentiel les activités effectuées par les différents acteurs humains du télé-TP.

Pour atteindre ces objectifs, le gestionnaire s'appuie sur un ensemble de trois composants : un référentiel renferme la connaissance de l'ensemble des laboratoires (locaux et partagés), un *Proxy Provider* WBEM interagit directement avec les ressources des laboratoires locaux, alors qu'un *Provider* WBEM dédié communique avec les serveurs WBEM secondaires responsables des laboratoires partagés.

6.2.2.2. Le référentiel

Le référentiel contient les informations de supervision décrivant l'ensemble des entités mises en jeu dans notre architecture globale : les environnements d'apprentissage, les laboratoires distants, et les utilisateurs interagissant avec ces deux types de systèmes. Pour représenter ces informations, nous avons élaboré des méta-modèles génériques de sorte à pouvoir les étendre pour prendre en charge les spécificités de chaque environnement d'apprentissage et d'expérimentation. Les détails de ces méta-modèles sont présentés dans le chapitre suivant mais, de façon succincte, ils comprennent :

- L'état d'allocation global des ressources de l'ensemble des laboratoires distants.
- L'état des expériences déployées, qu'elles soient dans l'état opérationnel, suspendu ou arrêté.
- Les traces d'activités réalisées par les utilisateurs à partir des EIAHs existants (et plus précisément à partir de l'IHM de télé-opération) sur les expériences ; ces traces comprennent les commandes formulées et les réponses associées, mais également les notifications d'évènements produites par les Pilotes embarqués dans les ressources des laboratoires.

La Figure 6-4 complète la Figure 5-9 de la section 5.3 et illustre la puissance et l'intérêt d'un référentiel centralisé situé au niveau de la couche d'Intégration. En effet, à la différence de la couche d'Expérimentation dont les Pilotes manipulent uniquement les classes relatives à l'instrumentation du dispositif qu'ils gèrent, la couche d'Intégration possède une vue globale de tous les dispositifs physiques ou virtuels impliqués dans une expérience, qu'elle soit centralisée ou distribuée. Ainsi, les applications de la couche supérieure peuvent aisément manipuler l'expérience dans sa globalité, sans rentrer dans les détails de chaque composant ni de leur localisation. D'autre part, les utilisateurs manipulent des objets graphiques grâce aux IHMs qui leur sont offertes, la couche d'Apprentissage manipule des modèles en format XML appropriés aux applications WEB, alors que les couches d'Intégration et d'Expérimentation manipulent les mêmes modèles en format natif MOF appropriés aux programmes de bas niveau.

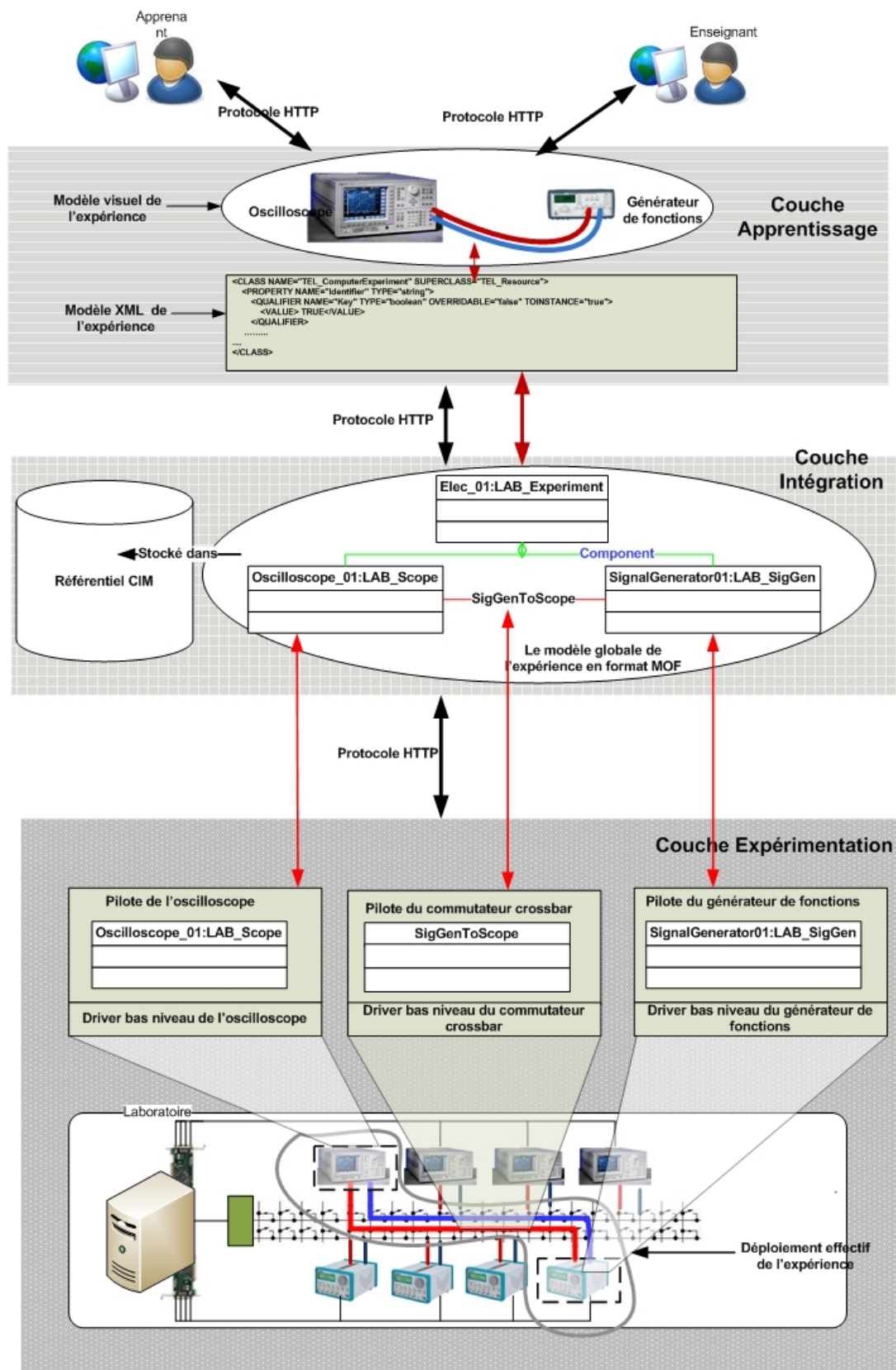


Figure 6-4. Exploitation des modèles d'expérience dans les différentes couches.

Le gestionnaire assure la consistance des données puisqu'il est le seul à pouvoir insérer ou mettre à jour des informations dans le référentiel. Toutefois, il est capable de déclencher une erreur lorsqu'une incohérence est détectée (tentative de duplication d'une donnée existante, privilèges insuffisants, etc.).

6.2.2.3. Le *Proxy Provider* WBEM

Le *Proxy Provider* WBEM représente un intermédiaire par lequel le gestionnaire primaire assure toutes les fonctions relatives aux expériences centralisées, car il communique avec les Pilotes embarqués dans les ressources des laboratoires locaux. Il peut alors transmettre des instructions émises par le gestionnaire et qui seront opérées par les Pilotes sur les ressources du laboratoire, demander certaines informations d'observation pour les stocker dans le référentiel via le gestionnaire, ou recevoir des notifications générées par les ressources du laboratoire et qui seront transmises au gestionnaire en vue de leur traitement.

6.2.2.4. Le *Provider* WBEM

L'objectif du *provider* WBEM est de faciliter la gestion des expériences distribuées sur des laboratoires délocalisés. Le *provider* WBEM communique avec les gestionnaires des serveurs secondaires qui, à leur tour, communiquent avec les *Proxy Providers* capables de piloter les ressources de leur(s) laboratoire(s) via les Pilotes embarqués. Une requête issue d'un service du niveau supérieur de la couche d'Intégration demandant le déploiement d'une expérience nécessitant des ressources distribuées sur différents laboratoires est décomposée, au niveau du gestionnaire primaire, en plusieurs requêtes destinées à différents serveurs WBEM secondaires. C'est le *Provider* qui se chargera de l'émission des différentes requêtes aux serveurs secondaires, et qui retournera au gestionnaire primaire les réponses rendues par les différents dispositifs de laboratoires.

6.2.3. Les serveurs WBEM secondaires

Les serveurs WBEM secondaires sont des composants optionnels dans notre architecture, mais indispensables pour atteindre les objectifs de fédération et de mutualisation des ressources partagées par différentes institutions. Ils implémentent une partie des fonctions assurées par les serveurs primaires, et notamment tous les processus de gestion, déploiement

et exploitation des ressources de leurs laboratoires. Un serveur WBEM secondaire est constitué des mêmes composants qu'un serveur primaire, à l'exception du *Provider* puisqu'un serveur secondaire n'a pas à gérer de laboratoires de plus bas niveaux. Par conséquent, un référentiel secondaire ne contient que des informations relatives à son laboratoire local, et n'offre donc qu'une vue partielle de l'état d'une expérience distribuée. De plus, il ne détient aucune donnée relative aux traces d'activités réalisées par les utilisateurs.

Les serveurs WBEM secondaires sont régulièrement interrogés par le gestionnaire primaire afin d'obtenir l'état des ressources des laboratoires qu'ils gèrent, et constituer ou mettre à jour l'état global de toutes les ressources partagées. Grâce à ces informations, stockées et mises à jour dans le référentiel primaire, il est possible de répondre aux requêtes de déploiement, de suspension et de libération des expériences sur les laboratoires distants partagés. Lors des différentes opérations de réservation, d'allocation et de libération de ressources partagées, le gestionnaire primaire modifie également l'état d'allocation des ressources au niveau des serveurs WBEM secondaires afin que tous les serveurs partagent une vue cohérente.

6.2.4. Les Pilotes des dispositifs de laboratoires

Il s'agit du composant décrit dans le chapitre 5.3. Un Pilote de dispositif est un composant de contrôle qui assure l'interaction entre un serveur WBEM (qu'il soit primaire ou secondaire) et une ou plusieurs ressources de laboratoire gérées par ce serveur. Un Pilote peut être embarqué directement dans la ressource gérée si cette dernière possède un système informatique capable de l'exécuter, autrement il doit être exécuté par un autre système de médiatisation d'une ou de plusieurs ressources.

Les Pilotes représentent donc le dernier maillon de la chaîne de contrôle reliant l'interface utilisateur à la ressource gérée, puisqu'ils sont capables de communiquer avec le serveur WBEM pour recevoir les commandes qu'ils vont traduire en routines de bas niveau compréhensibles par la ressource gérée, ou recevoir les réponses et événements provenant de cette dernière et les traduire en structures de données compréhensibles par le serveur WBEM.

6.2.5. Le protocole d'échange d'informations

Le protocole *CIM-XML* défini dans l'architecture WBEM est un protocole de communication de niveau applicatif, encapsulé dans le protocole HTTP, qui permet l'échange d'informations

de gestion dans le format standard *xmlCIM* ; ce format permet de représenter les données contenues dans le référentiel avec le langage XML, et ainsi de promouvoir le développement d'applications web de gestion qui exploitent efficacement ce langage.

Le protocole spécifie le format des messages échangés, et comprend des requêtes formulées par les clients ou gestionnaires WBEM, et des réponses retournées par les (*Proxy*) *Providers*. Il assure donc la communication entre ces entités, mais également entre les *Proxy Providers* et les Pilotes de la couche d'Expérimentation embarqués dans les ressources des laboratoires distants.

6.3. Synthèse

Notre choix de concevoir le niveau supérieur de la couche d'Intégration par une architecture orientée services apporte de multiples atouts :

- La spécification d'interfaces indépendantes des laboratoires visés permet d'uniformiser l'interaction avec les expériences.
- L'homogénéisation de la communication entre environnements d'apprentissage et laboratoires en ligne, puisqu'un unique protocole fondé sur les technologies de l'Internet assure les interactions entre ces deux types de systèmes.
- La simplification du processus de développement d'outils, services et IHMs pédagogiques qui sont nécessaires à une exploitation optimale des laboratoires distants et des expériences qu'ils hébergent. En effet, les développeurs n'ont pas à se soucier des détails techniques de bas niveau relatifs aux dispositifs physiques ou virtuels des laboratoires et peuvent se focaliser sur le développement d'interfaces génériques réutilisables par d'autres plateformes d'apprentissage et par d'autres acteurs humains.

D'autre part, le niveau inférieur fondé sur l'architecture WBEM satisfait plusieurs objectifs fixés précédemment :

- La création d'une vue homogène de l'ensemble des ressources gérées, et ce quelque soit leurs nature et localisation.
- Les EIAHs ont l'opportunité de déployer et d'exploiter des expériences distribuées sur plusieurs sites, puisque la hiérarchie des serveurs WBEM assure la fédération et la mutualisation de plusieurs laboratoires délocalisés.

- La majorité des dispositifs de laboratoires actuels peuvent être facilement pris en compte par la couche d'Intégration puisqu'ils intègrent, de plus en plus, des systèmes informatiques embarqués assurant leur télé-instrumentation. Notre approche considère donc la majorité des disciplines scientifiques et technologiques, seules les IHMs spécifiques à des dispositifs particuliers ainsi que les Pilotes correspondant à ces derniers doivent être développés.
- D'une manière générale, en optant pour le standard de gestion WBEM, nous bénéficions de la maturité de cette architecture et de sa capacité à gérer n'importe quelle entité physique ou logique, mais également de sa pérennité, de son évolutivité et des nombreux outils implémentant ce standard.

Chapitre 7. Des modèles génériques pour les télé-TPs

<u>7.1.</u>	<u>La modélisation CIM</u>	157
<u>7.1.1.</u>	<u>Le méta-modèle CIM</u>	157
<u>7.1.2.</u>	<u>Le modèle d'informations CIM</u>	160
<u>7.1.3.</u>	<u>CIM comme méta-modèle pour les OPI « Expérience »</u>	162
<u>7.2.</u>	<u>Modélisation du domaine des télé-TPs</u>	162
<u>7.2.1.</u>	<u>Sous-modèle de gestion des laboratoires</u>	163
<u>7.2.2.</u>	<u>Sous-modèle de gestion des expériences</u>	167
<u>7.2.3.</u>	<u>Exemple de modélisation d'une expérience</u>	170
<u>7.3.</u>	<u>Modélisation du domaine des EIAHs</u>	171
<u>7.4.</u>	<u>Modélisation des utilisateurs et de leurs activités</u>	175
<u>7.4.1.</u>	<u>Le profil de l'apprenant</u>	175
<u>7.4.2.</u>	<u>La représentation des activités</u>	177
<u>7.5.</u>	<u>Synthèse</u>	179

Dans le chapitre précédent, nous avons mis en évidence le nombre important d'informations nécessaires à la couche d'Intégration pour assurer les processus de gestion des laboratoires, des expériences et des traces d'activités. Nous présentons dans ce chapitre un ensemble de modèles qui correspondent aux informations décrites dans la base de connaissances associée au gestionnaire de la couche d'Intégration, et qui offrent une vue homogène des différentes entités physiques, logiques et humaines de notre contexte d'étude.

Le DMTF a élaboré un méta-modèle d'informations, le Common Information Model (CIM), pour décrire les réseaux et systèmes supervisés au travers de l'architecture WBEM. Le nouveau schéma « *LAB* » que nous avons introduit repose donc sur le modèle de gestion des ressources du DMTF (DMTF, 2009) pour représenter les laboratoires, ressources et expériences en ligne intégrées dans une activité de télé-TP, et nous étendons le modèle de gestion des traces de (Broisin et Vidal, 2007) pour observer les activités des utilisateurs sur les expériences.

7.1. La modélisation CIM

7.1.1. Le méta-modèle CIM

Le méta-modèle CIM permet d'unifier la représentation d'équipements, de services et d'applications hétérogènes, et ainsi de faciliter l'échange d'informations de gestion indépendamment des équipements supervisés. Issu du paradigme objet, CIM offre non seulement une vue conceptuelle de haut niveau sur les objets gérés, mais il unifie également tous les autres standards de gestion existants comme SNMP ou CMIP (DMTF, 2003). Les objets gérés sont modélisés par des classes qui encapsulent les propriétés et méthodes (au sens de la Programmation Orientée Objet), et CIM offre la possibilité d'établir des relations entre ces objets :

- La relation d'héritage et les propriétés de polymorphisme ascendant et descendant permettent de voir les objets CIM sous plusieurs angles selon les objectifs de l'application d'administration.
- Les relations d'association, de composition et d'agrégation établissent des liens sémantiques entre les objets gérés, et identifient les objets qui dépendent ou influent sur d'autres objets.

Les classes et instances des objets CIM sont exprimées par le langage MOF (*Managed Object Format*) fondé sur le langage IDL³⁶, alors que les modèles CIM sont visualisés sous la forme de diagrammes UML³⁷ (DMTF, 2009). Pour faciliter la lecture de diagrammes CIM-UML, des conventions de couleur facilitent la distinction des différentes relations : l'héritage en bleu, les associations en rouge, et les agrégations en vert. La Figure 7-1 illustre par un diagramme UML les éléments fondamentaux de CIM (ou le méta-méta-modèle CIM) exploitant ces conventions de couleurs. Notons que les classes et instances CIM peuvent également être exprimées par le langage XML (DMTF, 2007)³⁸ qui est utilisé pour représenter les données CIM échangées sur le réseau (entre un client et un gestionnaire

³⁶ Interface Definition Language

³⁷ Unified Modeling Language

³⁸ <http://www.dmtf.org/sites/default/files/standards/documents/DSP201.html>

WBEM, ou entre un gestionnaire et un *Provider*). Ce langage est formalisé par une DTD³⁹ que les applications de haut niveau peuvent alors exploiter.

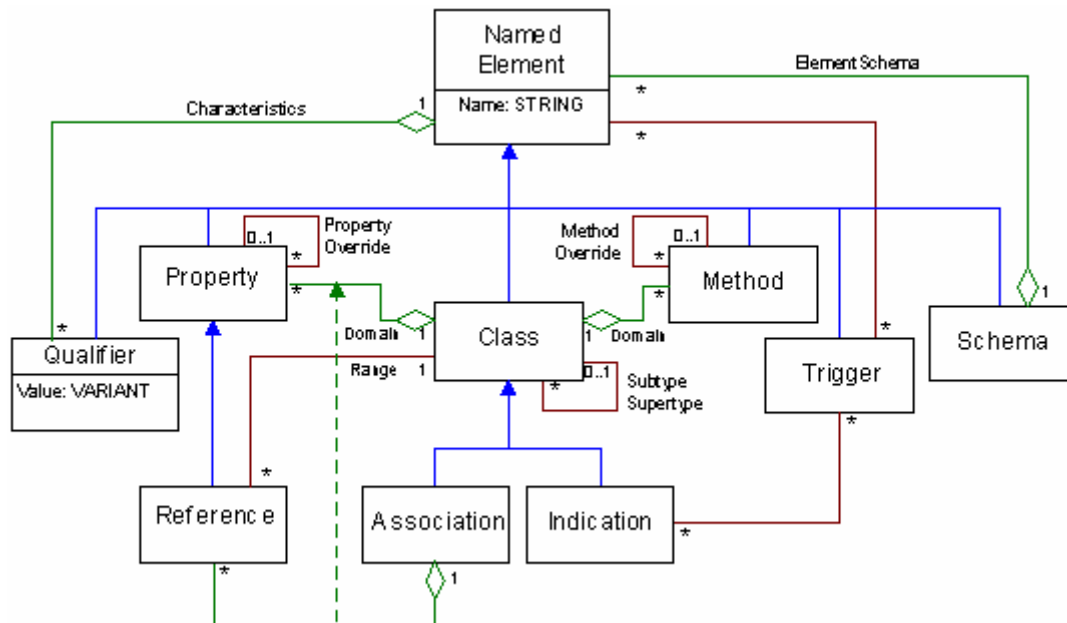


Figure 7-1. Méta-méta-modèle CIM illustré par un diagramme UML.(DMTF, 2003)

Les détails des éléments de modélisation figurent dans (DMTF, 2003) ou (Festor et Benyoussef, 2000), mais les entités suivantes nous intéressent particulièrement :

- Un *Schema* regroupe un ensemble de classes appartenant à un propriétaire unique, et dénote un domaine d'administration particulier. Par exemple, toutes les classes CIM proposées par le DMTF sont précédées du préfixe « CIM_ », alors que notre schéma conçu pour le domaine de l'EIAH utilise le préfixe « TEL⁴⁰_ ». D'une manière générale, le nom d'une classe doit être de la forme *SchemaName_ClassName*. Toutefois, pour des raisons de lisibilité, les classes natives CIM illustrées dans les figures de ce chapitre ne comportent pas le préfixe « CIM_ ».
- Une *Class* est l'élément principal du méta-modèle CIM. Elle est constituée de propriétés et de méthodes. Les associations, les agrégations et les indications ne sont que des classes particulières.

³⁹ En XML, DTD est l'acronyme de Document Type Definition

⁴⁰ Technology Enhanced Learning

- Une *Property* dénote une caractéristique d'une classe. Une propriété (ou attribut) possède un nom, un type de donnée, et optionnellement une valeur par défaut. Le type d'une propriété est limité aux types de données simples spécifiées dans la spécification CIM (DMTF, 2003), et aux tableaux de ces éléments. Un ou plusieurs attributs clés constituent ensemble l'identifiant d'un objet CIM, et le distinguent des autres instances de la même classe.
- Une *Method* est une opération qui peut être invoquée sur un objet CIM. Le traitement associé à une méthode peut se limiter à la mise à jour du référentiel ou engendrer l'exécution d'instructions sur le système géré dans le monde réel ; dans ce cas, la méthode doit être implémentée par le composant de contrôle responsable du système géré. La portée d'une méthode est limitée à la classe dans laquelle elle est définie, et sa signature comprend un nom, un type de retour, et des paramètres optionnels d'entrée et/ou de sortie. Grâce à ce concept, les opérations de gestion dépassent les fonctions primitives de lecture et d'écriture sur les variables de gestion proposées dans l'approche SNMP par exemple.
- Une *Association* est une classe particulière qui contient au moins deux références vers d'autres classes. Les associations n'affectent pas les classes qu'elles référencent, et ne peuvent hériter que des classes de même type.
- Une *Indication* est la représentation d'un événement associé à un déclencheur (*Trigger*).
- Un *Qualifier* ajoute des informations sémantiques aux éléments CIM. Il établit des contraintes comme les cardinalités des relations d'association.

Dans la Figure 7-2, nous illustrons un exemple de classe conforme à la modélisation CIM exprimée en trois langages : MOF (à gauche), UML (à droite) et XML (en bas). La classe *MyService* hérite de la classe parente *CIM_Service* définie par le DMTF, et spécifie les caractéristiques d'un service particulier. Les mots-clés *Key* et *MaxLen* correspondent à des *qualifiers* décrivant la propriété *Caption* de type chaîne de caractères ; ils indiquent respectivement que cette propriété est l'identifiant de la classe, et qu'elle est restreinte à une longueur de 256 caractères. Les méthodes *StartService()* et *StopService()* mettent à jour la propriété *Status*, mais aussi ordonnent le démarrage ou l'arrêt du service à son Pilote.



Figure 7-2. Exemple de classe CIM représentée aux formats MOF, UML et XML.

7.1.2. Le modèle d'informations CIM

Le modèle d'informations CIM repose sur une hiérarchisation de la représentation des objets gérés, et propose des modèles caractérisés par différents niveaux d'abstraction.

Le schéma CIM de base (*Core model*) contient un nombre minimal de classes et de relations qui décrivent d'une façon générique et abstraite tout type d'élément gérable (classe *ManagedSystemElement*), qu'il soit physique (classe *PhysicalElement*) ou logique (classe *LogicalElement*), et qui constitue les briques de base pour définir toute autre classe du modèle d'informations. La Figure 7-3 résume ce schéma sous la forme d'un diagramme UML.

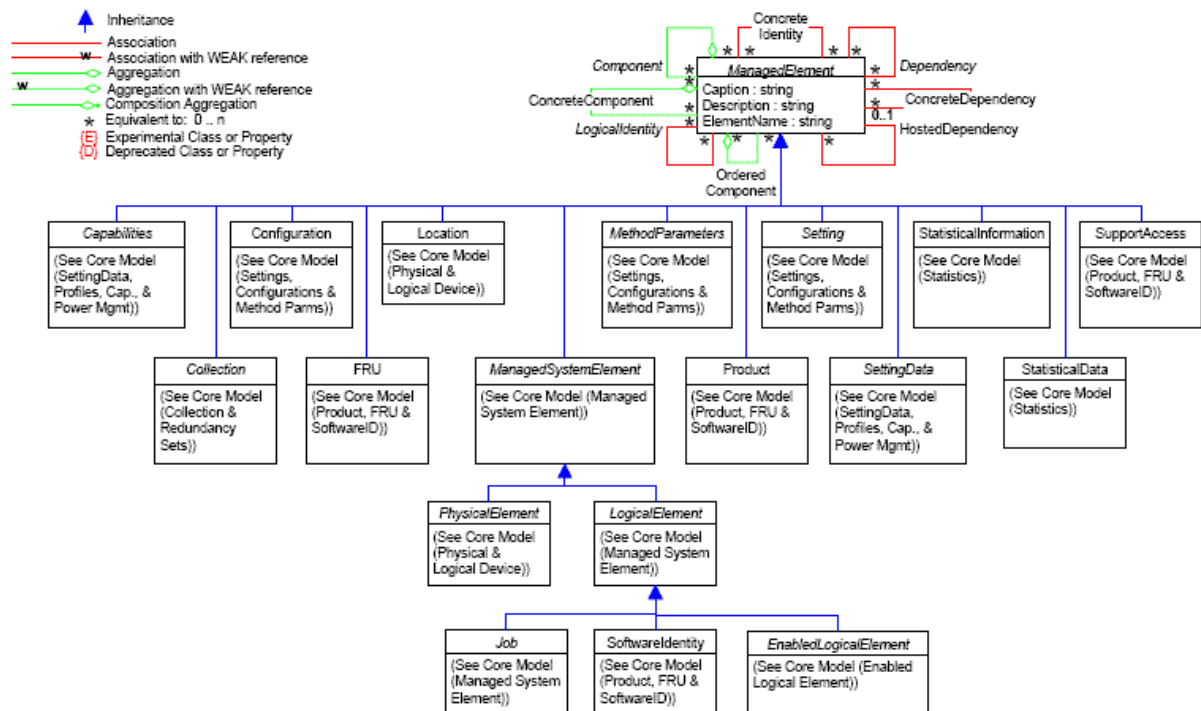


Figure 7-3. Le schéma CIM de base (DMTF, 2003).

Les schémas communs (*Common models*) étendent le schéma de base pour modéliser des domaines particuliers indépendamment de toute technologie particulière. Douze modèles communs sont proposés par le DMTF (DMTF, 2003), et présentés dans la Figure 7-4 : *Applications*, *Events*, *Network*, *Support*, *Database*, *Interoperability*, *Physical*, *Systems*, *Devices*, *Metrics*, *Policy* et *User*.



Figure 7-4. Les modèles communs de CIM (DMTF, 2003).

Lorsque les modèles communs ne suffisent pas pour décrire des aspects très spécifiques à un domaine ou système géré, ils sont étendus et spécialisés à travers les modèles d'extension

(*Extension models*). Comme exemple d'extension, nous pouvons citer l'ensemble des classes WIN32 définies par Microsoft™ dans le cadre spécifique de ses systèmes d'exploitation.

7.1.3. CIM comme méta-modèle pour les OPI « Expérience »

Comme nous l'avons mentionné dans la section 5.2, le format *xmlCIM* est celui préconisé au niveau de la couche d'Apprentissage pour décrire et exploiter l'Objet Pédagogique Interactif « Expérience ». La DTD correspondante peut être injectée dans un éditeur XML existant afin de manipuler aisément les modèles CIM.

D'autre part, le format CIM-OWL peut également être utilisé puisqu'il représente les modèles CIM sous la forme d'ontologies web. Cette représentation facilite l'exploitation des modèles d'expériences par les applications du web sémantique comme Protégé (Protégé, 2011), mais assure également la convergence avec les propositions de (Benmohamed, 2007) et (Gravier, 2007) présentées dans le chapitre 3 pour piloter les expériences à travers un moniteur IMS-LD générique comme Reload (Reload, 2006) ou CopperCore (CopperCore, 2008).

Ces différents formats de représentation du méta-modèle CIM renforcent notre choix. En effet, les modèles d'expériences, qu'ils soient présentés aux formats MOF, UML, *xmlCIM* ou CIM-OWL, sont des objets pédagogiques descriptibles par une fiche de métadonnées LOM qui peuvent être stockés dans un vivier d'objets pédagogiques.

Alors à partir des éléments de modélisation CIM et en suivant les recommandations du DMTF pour la conception de nouveaux modèles, nous avons élaboré un modèle générique dédié au domaine des télé-TPs. Ce modèle est partiellement présenté dans la section qui suit car il dépend du domaine d'enseignement, mais un modèle spécifique à l'apprentissage des systèmes et réseaux informatiques est détaillé dans la dernière partie de ce manuscrit.

7.2. Modélisation du domaine des télé-TPs

La Figure 7-5 illustre le modèle générique que nous avons conçu pour la description, le contrôle et la supervision des laboratoires et expériences en ligne, et se compose donc de deux parties : la gestion des laboratoires et la gestion des expériences. Il s'inspire essentiellement du modèle d'allocation de ressources proposé par le DMTF (DMTF, 2009) pour gérer tous types de ressources (ressources simples, hiérarchisées, virtuelles, etc.) à travers les différentes

phases de leur cycle de vie (allocation, libération, priorités, etc.). Vu la complexité de cette proposition, nous en avons repris les éléments essentiels pour implémenter une politique d'allocation simple qui peut être étendue pour satisfaire des objectifs plus complexes. Ainsi, nous avons introduit diverses classes et associations pour être en mesure de gérer non seulement les laboratoires et leurs ressources (*LAB_Domain*, *LAB_ResourcePool* et *LAB_Resource*), mais également les expériences qui sont associées aux utilisateurs autorisés à les exploiter (*LAB_Experiment*, *LAB_AuthorizedAccess*).

Figure 7-5. Modèle d'allocation des ressources de laboratoires pour les expériences en ligne.

Le modèle de gestion des laboratoires consiste à décrire ses caractéristiques d'un point de vue supervision, et à exprimer le fait qu'il comprend différents dispositifs physiques et/ou logiques qui eux aussi doivent être décrits d'un point de vue observation.

7.2.1.1. La classe *LAB_Domain*

Nous avons introduit la classe *LAB_Domain* dans notre nouveau domaine de gestion baptisé *LAB* pour modéliser un laboratoire qui héberge un ensemble (ou *pool*) de ressources. *LAB_Domain* hérite directement de la classe native *AdminDomain* définie comme étant un « *regroupement spécial de systèmes gérés, où l'ensemble est considéré comme une seule entité reflétant le fait que ses composants sont gérés par le même utilisateur, le même groupe ou la même politique et sert de point d'agrégation associant un ou plusieurs éléments gérés* » (DMTF, 2010a). Le Tableau 7-1 présente la liste des attributs d'un objet CIM de type *LAB_Domain*.

Tableau 7-1. Attributs de la classe *LAB_Domain*.

Identifiant	Code utilisé pour identifier d'une façon unique le laboratoire
<i>Localisation</i>	Pays, ville.
<i>Institution</i>	Nom de l'institution propriétaire du laboratoire.
<i>WebSite</i>	Site web de l'institution.
<i>Latitude</i>	Valeur de la latitude géographique.
<i>Longitude</i>	Valeur de la longitude géographique.
<i>AdminContact</i>	Nom du responsable administratif du laboratoire.
<i>EmailOfAdminContact</i>	Email du responsable administratif du laboratoire.
<i>TechContact</i>	Nom du responsable technique du laboratoire.
<i>EmailOfTechContact</i>	Email du responsable technique du laboratoire.
<i>Others</i>	Autres informations.

Notons que la classe *LAB_Domain* ne contient pas de méthode car elle décrit une entité administrative qui n'en a pas besoin. Une instance de cette classe permet d'atteindre les instances des classes qui décrivent les dispositifs du laboratoire appartenant à l'entité administrative en parcourant la relation de composition *Component*.

7.2.1.2. La classe *LAB_Resource*

Pour décrire un dispositif du laboratoire (ou ressource), et dans un objectif de généricité, nous avons réutilisé la classe native *ManagedSystemElement* issue du schéma *CIM Core*. Une ressource d'un laboratoire correspond alors à un élément physique (classe *PhysicalElement*) ou logique (classe *LogicalElement*) décrit par les attributs définis par le DMTF ; le lecteur est

invité à consulter la documentation CIM pour de plus amples détails (DMTF, 2011). Nous avons cependant introduit une nouvelle classe appelée *LAB_Resource* qui définit les propriétés nécessaires à la gestion de l'état d'utilisation des ressources dans le cadre d'expériences en ligne : chaque instance de cette classe est liée à l'instance de la ressource réelle par l'association *ElementAllocatedFromPool* qui exprime leur dépendance existentielle. Le Tableau 7-2 présente les attributs et méthodes de la classe *LAB_Resource* :

- Les attributs *IsFree* et *ExperimentAllocationID* indiquent respectivement l'état d'allocation de la ressource et l'identifiant de l'expérience à laquelle elle est rattachée. A titre d'exemple, la Figure 7-5 illustre trois ressources fréquemment rencontrées dans les laboratoires en ligne : les dispositifs physiques (*LAB_Device*), les logiciels (*LAB_SoftwareIdentity*), et les services (*LAB_Service*).
- La méthode *Reserve()* assure le blocage d'une ressource pour l'allouer à une expérience particulière.
- La méthode *Init()* est invoquée lors du déploiement de l'expérience qui va utiliser la ressource afin de lancer les opérations d'initialisation nécessaires au démarrage de l'expérience.
- La méthode *Free()* sert à déclarer la ressource comme étant libre à la fin d'une période d'expérimentation.

Tableau 7-2. Attributs additionnels de la classe *LAB_Resource*.

Attribut	Valeur	Remarques
<i>IsFree</i>	<i>TRUE/FALSE</i>	<i>TRUE</i> : ressource disponible. <i>FALSE</i> : ressource indisponible.
<i>ExperimentAllocationID</i>	<i>SystemID:ExperimentID/NULL</i>	<i>SystemID</i> : identifiant du système qui héberge la ressource (adresse IP ou nom de domaine). <i>ExperimentID</i> : identifiant de l'expérience.
Méthode	Résultat	Remarques
<i>Reserve()</i>	<i>Retourne 0 si l'opération réussie, 1 sinon.</i>	Assure que la ressource est libre, puis fixe l'attribut <i>IsFree</i> à la valeur <i>False</i> et l'attribut <i>ExperimentAllocationID</i> avec l'identifiant de l'expérience.
<i>Init()</i>	<i>Retourne 0 si l'opération réussie, 1 sinon.</i>	Initialise les attributs de l'objet qui décrit la ressource.
<i>Free()</i>	<i>Retourne 0 si l'opération</i>	Fixe l'attribut <i>IsFree</i> à la valeur <i>True</i> et

	<i>réussie, 1 sinon.</i>	<i>ExperimentAllocationID</i> à <i>NULL</i> .
--	--------------------------	---

7.2.1.3. La classe *LAB_ResourcePool*

Nous avons introduit la classe *LAB_ResourcePool*, dont les attributs hérités de la classe native *ResourcePool* sont présentés dans le Tableau 7-3, pour attribuer une ressource à un laboratoire. Cette classe permet en effet de modéliser un ensemble de ressources à travers la relation d'agrégation native *Component*, qui est finalement rattaché à un laboratoire grâce à la relation *HostedResourcePool*. Cette modélisation permet un découpage fin et précis de la gestion des ressources d'un laboratoire qui offre aux institutions l'opportunité de mettre en œuvre des politiques d'allocation diverses. Par exemple, si une institution ne souhaite partager ses ressources avec aucune organisation, alors une politique d'allocation peut consister à agréger les ressources dans un unique *pool* hébergé par le laboratoire ; si, par contre, un organisme souhaite mettre à disposition l'ensemble ou une partie de ses ressources à différents partenaires, alors une autre politique serait de constituer différents *pools* de ressources, chacun hébergé par le partenaire bénéficiaire du *pool*. Enfin, l'association *ElementAllocatedFromPool* permet de retrouver, à partir d'un *pool* donné, les détails des différentes ressources qui le composent.

Tableau 7-3. Attributs de la classe *LAB_ResourcePool*.

Attribut	Valeur
<i>InstanceID</i>	Identifiant unique du <i>pool</i> de ressources.
<i>Capacity</i>	Nombre total de ressources appartenant au <i>pool</i> .
<i>Reserved</i>	Nombre de ressources réservées.
<i>ResourceType</i>	Type de ressources regroupées dans le <i>pool</i> .
<i>OtherResourceType</i>	Type de la ressource défini par l'utilisateur, lorsqu'aucun type prédéfini n'est satisfaisant.
<i>ResourceSubType</i>	Valeur fournie lorsque le <i>pool</i> correspond à un sous-type.
<i>AllocationUnits</i>	Fixe l'unité minimum d'allocation (par exemple, pour l'allocation d'une unité de type mémoire vive, l'unité d'allocation minimum est de 10000 octets).

7.2.2. Sous-modèle de gestion des expériences

Afin de représenter les expériences qui prennent place au sein des laboratoires, nous avons introduit deux classes : la première est dédiée à la modélisation des expériences elles-mêmes, alors que la deuxième assure les mécanismes d'autorisation des utilisateurs sur celles-ci.

7.2.2.1. La classe *LAB_Experiment*

La classe *LAB_Experiment* consiste en un domaine administratif selon les termes de la définition précédente ; par conséquent, elle hérite de la classe *AdminDomain*. *LAB_Experiment* définit les attributs additionnels suivants :

- *ExperimentID* identifie une expérience de manière unique.
- *Deployed* indique que l'expérience est déployée ou non.
- *Started* indique que l'expérience est démarrée ou non.
- *Distributed* indique que l'expérience est centralisée ou distribuée.

Une instance de cette classe expose différentes méthodes présentées dans le Tableau 7-4. Dans notre approche, en fonction de son caractère centralisé ou distribué, une expérience est constituée de ressources hébergées par un ou plusieurs laboratoires. Ainsi, les relations de composition *Component* ou *SystemComponent* permettent d'associer une ressource à une expérience. A partir des classes du modèle de gestion des laboratoires, et notamment la classe *LAB_ResourcePool*, il est alors possible de retrouver l'ensemble des laboratoires hébergeant les dispositifs constituant une expérience donnée.

Tableau 7-4. Méthodes de la classe *LAB_Experiment*.

Méthodes	Paramètres d'entrée/sortie	Remarques
<i>Create()</i>	Identifiant de l'expérience Modèle CIM de l'expérience	Effectue de manière chronologique les opérations suivantes : <ul style="list-style-type: none">• Affectation de l'identifiant de l'expérience (<i>ExperimentID</i>).• Parcours des associations <i>Component</i> ou <i>SystemComponent</i> pour identifier les ressources.• Parcours des associations <i>Component</i> des instances <i>LAB_Domain</i> pour rechercher les <i>LAB_ResourcePool</i> contenant les ressources requises.• Vérification de la disponibilité des ressources requises.• Parcours des associations <i>Component</i> des instances <i>LAB_ResourcePool</i> pour identifier les ressources

		<p>disponibles</p> <ul style="list-style-type: none"> • Sélection des ressources puis invocation, sur chaque ressource, de la méthode de réservation. • Création des instances de l'association <i>Component</i> ou <i>SystemComponent</i> entre l'instance de l'expérience et chaque ressource en fonction du modèle fourni en paramètre. • Initialisation des attributs de chaque ressource en fonction du modèle de l'expérience. • Affectation de la valeur <i>True</i> à l'attribut <i>Started</i>. • Affectation de la valeur <i>True</i> à l'attribut <i>Distributed</i> si les ressources de l'expérience sont distribuées. <p>Retourne 1 si l'une des opérations précédentes échoue, 0 sinon.</p>
<i>Start()</i>	Identifiant de l'expérience	<p>Effectue de manière chronologique les opérations suivantes :</p> <ul style="list-style-type: none"> • Vérification de la valeur <i>True</i> à l'attribut <i>Deployed</i>. • Parcours de toutes les ressources de l'expérience à travers les associations <i>Component</i> et <i>SystemComponent</i> pour invoquer la méthode d'initialisation de chaque ressource. <p>Retourne 1 si l'une des opérations précédentes échoue, 0 sinon</p>
<i>Stop()</i>	<p>Identifiant de l'expérience</p> <p>Modèle CIM dans son dernier état</p>	<p>Effectue de manière chronologique les opérations suivantes :</p> <ul style="list-style-type: none"> • Vérification de la valeur <i>True</i> à l'attribut <i>Started</i>. • Parcours de toutes les ressources à travers les associations <i>Component</i> et <i>SystemComponent</i> pour invoquer la méthode de libération de chaque ressource. • Parcours de toutes les instances <i>LAB_ResourcePool</i> associées à chaque ressource par l'association <i>Component</i> pour mettre à jour l'état d'allocation global du <i>pool</i>. • Suppression des associations <i>Component</i> ou <i>SystemComponent</i> qui relient l'expérience à ses ressources. <p>Retourne 1 si l'une des opérations précédentes échoue, 0 sinon.</p> <p>Dans le dernier cas, le modèle CIM de l'expérience dans son dernier état est retourné.</p>
<i>Resume()</i>	Identifiant de l'expérience	<p>Exécute les mêmes opérations que la méthode <i>Start()</i>, mais le modèle CIM fourni pour initialiser l'expérience correspond à son état avant l'arrêt de l'expérience.</p>

		Retourne 1 en cas d'échec, 0 sinon.
<i>Save()</i>	Identifiant de l'expérience	Prend un cliché de tout le modèle de l'expérience et le sauvegarde dans la base de connaissances. Retourne l'identifiant du modèle sauvegardé, 0 sinon.
<i>Restore()</i>	Identifiant de l'expérience Identifiant du modèle	Exécute les mêmes opérations que la méthode <i>Resume()</i> , mais redémarre l'expérience selon l'identifiant de sauvegarde fourni en argument. Retourne 1 en cas d'échec, 0 sinon.

Les méthodes exposées ci-dessus permettent de gérer le cycle de vie d'une expérience et correspondent aux différents services offerts à la couche d'Apprentissage pour l'administration des expériences. Toutefois, la phase de conception n'est pas prise en considération puisqu'elle s'effectue a priori par un (enseignant) concepteur, et consiste à produire le modèle CIM au format MOF correspondant à l'expérience conçue, appelé gabarit dans la section 5.2 ; il s'agit donc de l'objet pédagogique « Expérience » *a priori*.

La phase de déploiement, quant à elle, correspond à l'instanciation des différentes classes des dispositifs qui composent l'expérience ainsi que de leurs associations, et qui figurent dans le gabarit.

7.2.2.2. La classe *LAB_AuthorizedAccess*

Dans l'approche du DMTF, la classe *Identity* modélise un utilisateur et représente la brique de base des systèmes de sécurité pour offrir les services d'authentification et d'autorisation ; elle peut correspondre à un compte utilisateur dans un système UNIX ou dans un annuaire LDAP. Nous rappelons que dans notre contexte, les services d'authentification et d'autorisation sont assurés par un système externe à la couche d'Intégration, généralement un système d'apprentissage ou un service d'authentification unique (SSO⁴¹). Alors pour associer une expérience à un ou plusieurs utilisateurs, nous avons introduit la relation de dépendance *LAB_AuthorizedAccess* qui exprime le fait qu'un utilisateur a été authentifié, et qui autorise l'usager à exploiter l'expérience. L'attribut booléen *ReadWrite* affine les droits de l'utilisateur

⁴¹ Single Sing-On

sur l'expérience : lorsqu'il est positionné à *True*, l'utilisateur possède les droits d'accès en consultation et en modification, alors que la valeur *False* restreint l'accès en lecture sur l'expérience. Les instances de cette classe sont donc créées lorsqu'un utilisateur s'authentifie sur un des systèmes cités ci-dessus.

7.2.3. Exemple de modélisation d'une expérience

La Figure 7-6 illustre l'opérationnalisation de notre exemple d'expérience en génie électrique présenté tout au long du chapitre 5 à travers notre modèle de gestion des laboratoires et des expériences ; pour cela, nous utilisons un type particulier de diagramme UML, le diagramme d'instances (également appelé diagramme d'objets).

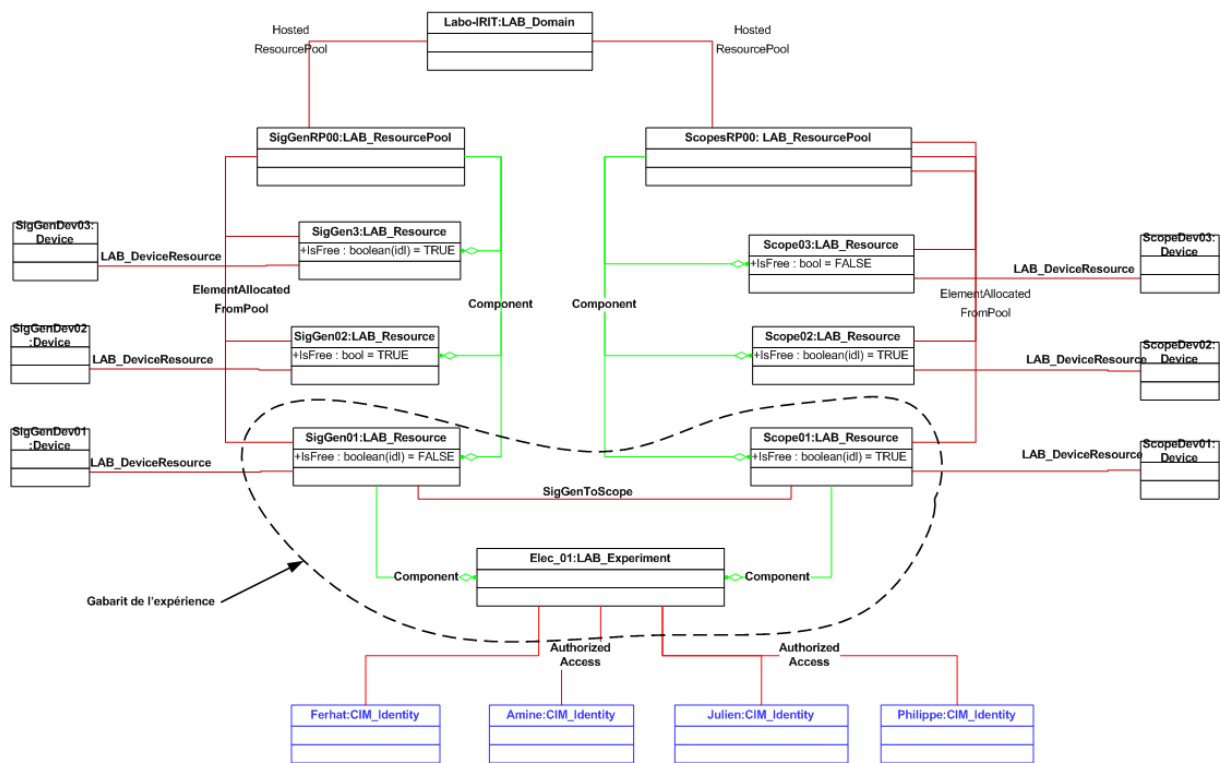


Figure 7-6. Diagramme d'objets illustrant l'exemple de l'expérience en génie électrique.

Dans cet exemple, les agrégations de type *Component* montrent qu'une expérience nommée *Elec_01* est constituée de deux instruments appartenant chacun à deux *pools* de ressources différents :

- La ressource *Scope01* qui appartient au *pool* *ScopesRP00* est associée à l'équipement *ScopeDev01* de type *Device* décrivant un oscilloscope.

- La ressource *SigGen01* qui appartient au *pool SigGenRP00* est associée à l'équipement *SigGenDev01* de type *Device* décrivant un générateur de fonctions.
- L'association *SigGenToScope* qui décrit la connexion entre les deux équipements est générée à partir du modèle gabarit de l'expérience.

Enfin, les associations *AuthorizedAcces* donnent le droit d'accès aux utilisateurs *Tuteur01*, *Apprenant01*, *Apprenant02* et *Apprenant03* sur cette expérience, et déterminent le niveau de privilège accordé (lecture seule, lecture et écriture).

Nous avons défini dans cette section un modèle générique pour la supervision des laboratoires et expériences en ligne qui assure la gestion du cycle de vie de ces expériences. Toutefois, nous avons montré dans le chapitre précédent que la couche d'Intégration nécessite la connaissance des informations d'observation qui décrivent les environnements d'apprentissage (voir section 6.1.4). Alors dans la section suivante, nous présentons un modèle décrivant les EIAHs d'un point de vue supervision.

7.3. Modélisation du domaine des EIAHs

L'objectif de la modélisation du domaine des EIAHs vise le suivi des activités des apprenants pour recueillir les traces d'interactions avec les différents objets pédagogiques, outils et systèmes d'apprentissage, et ainsi proposer divers services pédagogiques : suivi et évaluation de l'apprentissage, réingénierie pédagogique de scénarios d'apprentissage, réingénierie des systèmes d'apprentissage, analyse a posteriori de l'apprentissage, etc.

De plus, dans notre architecture définie dans les chapitres précédents, nous avons situé l'IHM de télé-opération au sein des EIAHs existants afin de masquer l'hétérogénéité des dispositifs de laboratoires, d'offrir un moyen de communication unique, etc. Alors pour être en mesure d'observer les actions des utilisateurs au sein de ces systèmes d'apprentissage, nous avons adopté l'approche dirigée par les modèles proposée par (Broisin et Vidal, 2007) qui s'intéresse à l'externalisation et au partage de traces d'activités entre différents systèmes, applications et acteurs.

Dans le contexte de ces travaux, un modèle caractérisé par un haut niveau d'abstraction a été élaboré pour représenter les systèmes et ressources spécifiques au domaine de l'apprentissage

en ligne (Butoianu et al., 10 ; Butoianu et al., 11). Nous nous appuyons donc sur ce modèle illustré par la Figure 7-7 pour représenter les EIAHs supports aux activités de télé-TP.

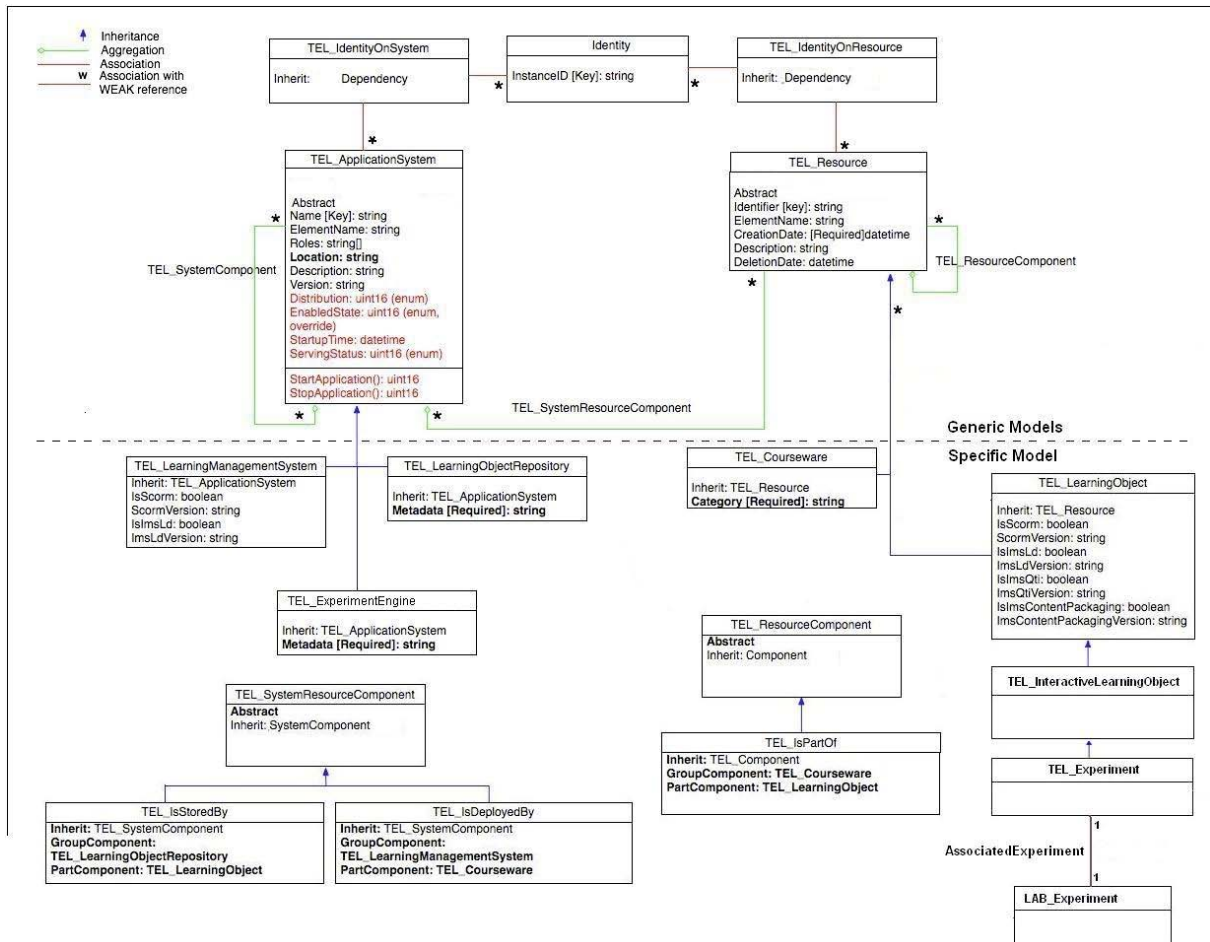


Figure 7-7. Modélisation du domaine des EIAHs.

La modélisation proposée pour décrire les systèmes et ressources d'un EIAH repose sur deux classes principales et trois relations d'agrégation. La classe abstraite *TEL_ApplicationSystem* spécifie les données décrivant les systèmes d'apprentissage, alors que la classe abstraite *TEL_Resource* décrit toute ressource intégrée dans un système d'apprentissage, et qui est disponible pour ce système. Ces classes héritent toutes deux de classes CIM prédéfinies dans les modèles natifs (respectivement *ApplicationSystem* et *Resource*), et bénéficient implicitement d'une relation d'agrégation qui a été spécialisée par l'association *TEL_SystemResourceComponent* pour identifier les différentes ressources constituant un système d'apprentissage. La relation de composition *TEL_SystemComponent* (ou *TEL_ResourceComponent*) entre des instances de la classe *TEL_ApplicationSystem* (ou *TEL_Resource*) permet d'exprimer le fait qu'un système (ou une ressource) d'apprentissage

peut être constitué(e) d'un(e) ou plusieurs autres systèmes (ou ressources). A partir de ce modèle abstrait, d'autres modèles plus spécifiques peuvent être définis pour décrire divers systèmes tels que des viviers de connaissances ou des plateformes d'apprentissage, ainsi que différents types de ressources comme les objets pédagogiques ou les cursus d'apprentissage ; l'illustration de ce modèle spécifique apparaît dans la partie inférieure de la Figure 7-7. D'autre part, les relations d'association *TEL_IdentityOnSystem* et *TEL_IdentityOnResource* permettent de distinguer les utilisateurs interagissant avec les systèmes et ressources d'apprentissage. En effet, ces associations référencent, respectivement, un système ou une ressource d'apprentissage d'un côté, et une identité physique de l'autre, modélisée par la classe *Identity*. Ces relations jouent un rôle clé pour associer une activité à une paire utilisateur/système ou utilisateur/ressource.

Alors pour représenter notre objet pédagogique « Expérience », nous avons introduit la classe *TEL_InteractiveLearningObject* dédiée aux OPIs et qui hérite de la classe *TEL_LearningObject*. L'OPI « Expérience » baptisé *TEL_Experiment* et détaillé par la Figure 7-8 exprime les propriétés pédagogiques d'une expérience à travers les attributs *Identifier*, *Description* et *Objectives*. Elle contient également la description technique détaillée *a priori* obtenue à la fin de la phase de conception puisque l'attribut *Template* est un URI⁴² qui pointe vers le fichier du modèle au format *cimXML*. Les méthodes définies dans cette classe correspondent à celles spécifiées dans la classe *LAB_Experiment* et peuvent être invoquées par le moteur d'exécution présenté dans la section 5.2.2 ; ce dernier est représenté sur la Figure 7-7 par la classe *TEL_ExperimentEngine* héritant de la classe *TEL_ApplicationSystem*.

⁴² Unified Resource Identifier

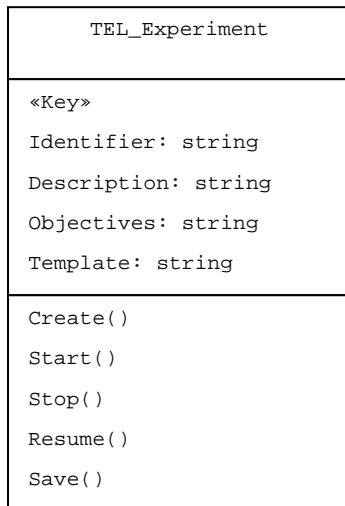


Figure 7-8. Modèle UML de la classe *TEL_Experiment*.

La classe *TEL_Experiment*, et en particulier l'attribut *Template*, représente le point de départ de l'instanciation de la classe *LAB_Experiment* : c'est en effet à partir du modèle a priori que vont être créées les différentes instances requises pour la mise en œuvre de l'expérience. La Figure 7-9 illustre, à partir de notre exemple d'expérience en génie électrique, la relation d'association *TEL_AssociatedExperiment* entre un objet pédagogique « Expérience » et une expérience déployée sur un laboratoire distant : au milieu l'instance de l'OPI « Expérience » (*Elec_01:TEL_Experiment*), à droite le modèle opérationnel correspondant (*Elec_01:LAB_Experiment*), et à gauche le modèle à priori au format *cimXML* pointé par l'attribut *Template*.

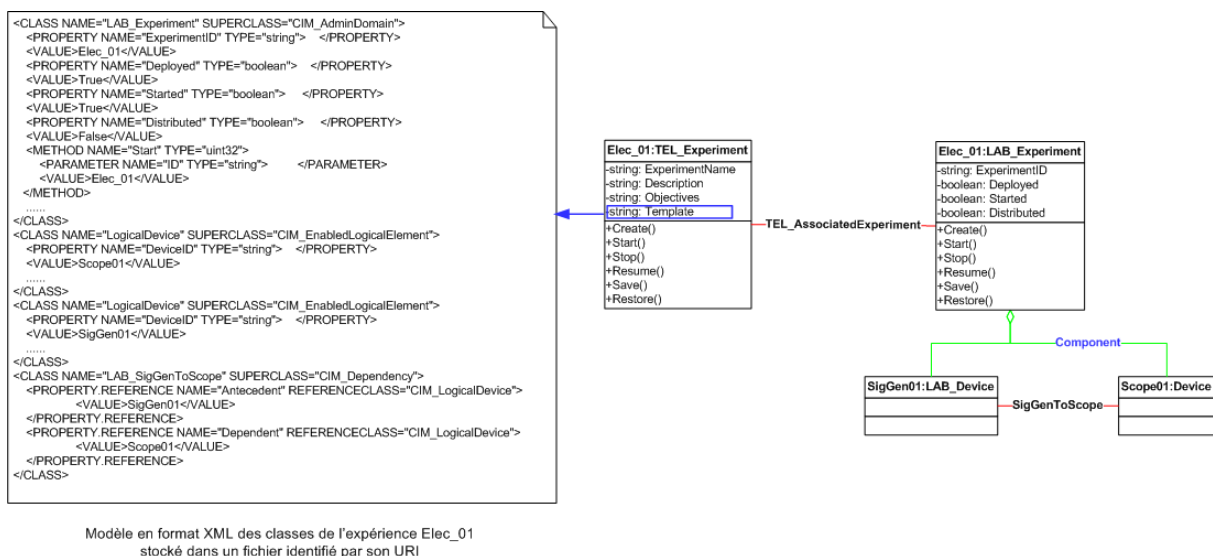


Figure 7-9. Association entre les classes *TEL_Experiment* et *LAB_Experiment*.

La spécification CIM d'une identité ne permet pas de décrire un utilisateur dans le contexte de l'apprentissage en ligne ; alors pour décrire plus finement un acteur de l'e-formation, la section qui suit s'intéresse à la modélisation de ces utilisateurs et des activités qu'ils peuvent réaliser sur les systèmes et ressources décrits plus haut.

7.4. Modélisation des utilisateurs et de leurs activités

Les modèles présentés dans cette section visent (1) la description des utilisateurs d'un point de vue pédagogique, en se focalisant sur les caractéristiques des apprenants, et (2) la représentation des activités, ou actions, qui peuvent être réalisées par les utilisateurs sur l'environnement d'apprentissage global.

7.4.1. Le profil de l'apprenant

Un profil de l'apprenant se définit comme un ensemble d'informations interprétées qui concernent un apprenant ou un groupe d'apprenants, collectées ou déduites à l'issue d'une ou plusieurs activités pédagogiques, qu'elles soient ou non informatisées. Les informations contenues dans le profil de l'apprenant peuvent concerner ses connaissances et compétences, ou encore son comportement (Jean-Daubias, 2005), et sont souvent exploitées pour permettre aux enseignants d'évaluer l'évolution des niveaux d'apprentissage (Martinez, 1989), ou d'adapter le scénario pédagogique selon les dispositions de chaque apprenant (Carron, 2006). Dans notre contexte, la description de l'apprenant sert les processus de tutorat et de collaboration, elle n'est pas destinée à être réutilisée dans un processus de personnalisation.

D'autres travaux de doctorat menés au sein de notre équipe s'intéressent à la mise en œuvre d'un système de tutorat intelligent, et proposent un profil de l'apprenant que nous avons adopté dans notre contexte des télé-TPs (Ramandalahy et al., 2009). L'approche consiste ici à exploiter et étendre les modèles CIM de base afin de prendre en compte les caractéristiques pédagogiques d'un utilisateur.

Le modèle *User* (DMTF, 2009b) propose trois classes pour représenter un utilisateur : la classe *Identity*, mentionnée précédemment, identifie un utilisateur dont les données administratives sont renseignées dans la classe *Person*, alors que des détails supplémentaires tels que le format de message électronique ou la langue préférés constituent la classe *OtherPersonInformation* (seuls quelques attributs apparaissent sur la Figure 7-10). Ce

modèle, même s'il spécifie un nombre important d'informations, ne décrit pas les utilisateurs d'un point de vue pédagogique. Alors en exploitant les capacités d'extension de ce modèle, de nouvelles classes illustrées sur la Figure 7-10 ont été spécifiées.

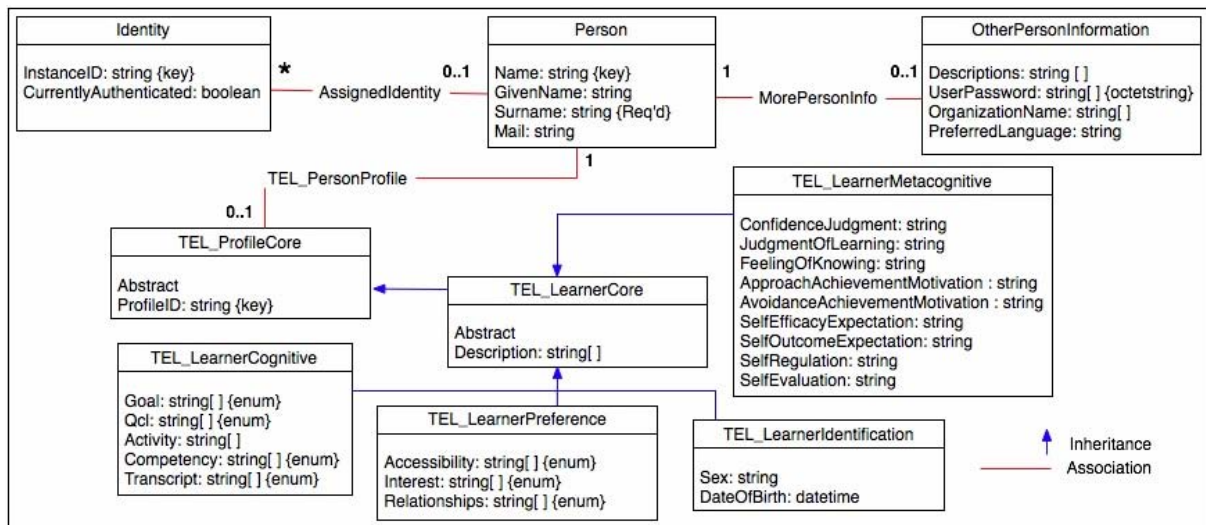


Figure 7-10. Le profil de l'apprenant.

La classe abstraite *TEL_ProfileCore* représente la classe racine du profil d'un utilisateur. Cette abstraction couvre les profils non spécifiques aux apprenants et permet de couvrir les différents rôles des utilisateurs dans un EIAH (enseignant, tuteur, administrateur, etc.). L'élément de plus haut niveau du profil de l'apprenant est représenté par la classe *TEL_LearnerCore* qui se décompose en quatre sous-profils :

- Le profil *TEL_LearnerIdentification* est relatif aux informations administratives de l'apprenant, et intègre les attributs définis dans la catégorie *Identification* du standard *Learner Information Package* (LIP).
- Le profil *TEL_LearnerCognitive* mesure les performances, buts et compétences de l'apprenant ; les autres catégories de LIP sont contenues dans ce sous-profil.
- Le profil *TEL_LearnerMetacognitive* mesure comment les apprenants réfléchissent à leurs propres connaissances.
- Le profil *TEL_LearnerPreference* inclut trois catégories de LIP et détaille les préférences et intérêts de l'apprenant.

Ce profil constitue une base pour décrire les caractéristiques pédagogiques d'un apprenant. Bien que son niveau d'abstraction permette d'étendre les classes proposées, voire même d'en

introduire de nouvelles, nous nous appuyons sur ce modèle de base pour modéliser les utilisateurs participant à une activité de télé-TP.

7.4.2. La représentation des activités

En complément de la modélisation des environnements et ressources d'apprentissage, les travaux de (Butoianu et al., 2010) proposent un modèle abstrait illustré par la Figure 7-11 pour représenter les activités des utilisateurs. Les classes « basses » de cette figure reprennent les principaux éléments exposés plus haut pour décrire les systèmes, ressources et utilisateurs. Les activités réalisées sur des systèmes sont modélisées par la classe abstraite *TEL_SystemActivity*, alors que *TEL_ResourceActivity* représente les actions effectuées sur les ressources. La notion de composition entre activités de même type est exprimée par les relations d'agrégation *TEL_SystemActivityComponent* et *TEL_ResourceActivityComponent*, alors que la composition d'activités de types différents est modélisée par la classe *TEL_SystemResourceActivityComponent*. Enfin, pour identifier une activité réalisée par un usager sur un système ou une ressource, les associations *TEL_DependencySystemActivity* et *TEL_DependencyResourceActivity* ont été introduites pour référencer une activité et une paire utilisateur/système ou utilisateur/ressource.

Nous avons spécialisé ce méta-modèle générique pour prendre en compte les spécificités des expériences en ligne. La Figure 7-11 représente le modèle générique de gestion des traces d'activités et l'extension que nous avons apportée.

[Liste_Arguments][Liste_Options] qui peuvent retourner un message de type *[Réponse]*. Alors pour représenter les actions exécutées via une ligne de commande, nous proposons la classe *TEL_CommandExperimentActivity* qui hérite de la classe *TEL_ExperimentActivity*. Chaque instance de cette classe contient la commande et, le cas échéant, la réponse du dispositif après exécution de cette commande, mais également les messages correspondant à la notification de l'occurrence d'un évènement.

Enfin, une session de travail, qui délimite les opérations exécutées par un utilisateur pour atteindre un objectif donné, est représentée par la classe *TEL_SessionExperimentActivity*. La relation de composition *TEL_ResourceActivityComponent* définie entre deux entités de la classe *TEL_ResourceActivity* est exploitée pour exprimer l'appartenance d'une trace d'activité (instance de la classe *TEL_CommandExperimentActivity*) à une session de travail (instance de la classe *TEL_SessionExperimentActivity*).

7.5. Synthèse

Nous avons mis en exergue dans la première partie de ce manuscrit des verrous relatifs à l'hétérogénéité des modèles de laboratoires, d'expériences, et de traces d'activité. Pour pallier ces manques, nous avons élaboré dans ce chapitre un modèle abstrait de haut niveau doté de la capacité d'intégration des modèles existants et futurs, et qui satisfait les besoins exprimés par la couche d'Intégration pour implémenter l'ensemble des mécanismes et services qu'elle propose. En effet, notre modèle est capable de décrire, indépendamment de toute technologie, un laboratoire, les ressources dont il dispose, ainsi que des expériences centralisées ou distribuées exploitant ces ressources de laboratoires.

D'autre part, nous avons étendu certains modèles existants élaborés dans notre équipe de recherche afin de décrire les systèmes et ressources d'apprentissage spécifiques aux télé-TPs. En particulier, nous avons proposé la modélisation de l'objet pédagogique interactif « Expérience », à partir des concepts que nous avons introduits dans le premier chapitre de cette partie, ainsi que les activités spécifiques au domaine des télé-TPs.

Dans la dernière partie du manuscrit, nous verrons comment ces modèles peuvent être spécialisés et opérationnalisés pour répondre aux besoins d'une discipline particulière qui est l'enseignement de l'informatique.

Chapitre 8. Synthèse des contributions

<u>8.1.</u>	<u>Du point de vue technique et technologique</u>	180
<u>8.1.1.</u>	<u>Une modélisation standardisée des télé-TPs</u>	181
<u>8.1.2.</u>	<u>Des interfaces de contrôle et de supervision uniformes</u>	183
<u>8.2.</u>	<u>Du point de vue pédagogique</u>	184
<u>8.2.1.</u>	<u>Intégration et interaction de l'objet « Expérience » avec les EIAHs</u>	184
<u>8.2.2.</u>	<u>Support à la conception d'expériences</u>	185
<u>8.2.3.</u>	<u>Support aux interactions avec les expériences et au tutorat synchrone</u>	185
<u>8.2.4.</u>	<u>Des IHMs adaptées</u>	185
<u>8.2.5.</u>	<u>Support de l'apprentissage collaboratif et du tutorat asynchrone</u>	186

Dans les trois chapitres précédents nous avons présenté une solution générique pour répondre à notre problématique générale de mise en ligne de travaux pratiques. Plus précisément, notre contribution vise à répondre aux contraintes énoncées dans le chapitre 4 qui synthétise, de notre point de vue, les lacunes identifiées dans les projets existants.

Rappelons que notre objectif principal, au-delà du déploiement de télé-TPs, est d'atteindre une efficacité pédagogique accrue tout en facilitant et promouvant l'accès à ce type d'activité à travers l'intégration des laboratoires dans les EIAHs existants. Dans ce chapitre, nous analysons notre solution selon les dimensions technique/technologique et pédagogique afin de déterminer dans quelle mesure elle contribue à atteindre ces objectifs.

8.1. Du point de vue technique et technologique

Nous analysons ici notre approche dans ses dimensions technique et technologique et mettons en avant leur caractère standardisé qui, selon notre point de vue, est indispensable à la bonne conduite des télé-TPs.

8.1.1. Une modélisation standardisée des télé-TPs

L'adoption de la modélisation fondée sur le méta-modèle CIM nous permet de décrire d'une façon homogène et uniforme tous les aspects des éléments d'un télé-TP, quelle que soit leur nature. Par rapport aux langages de modélisation des plateformes existantes présentés dans le chapitre 3, nous bénéficions de multiples intérêts :

- Simplicité et généricité : grâce à ses différents formats textuels MOF et XML, CIM est compréhensible et exploitable aussi bien par les humains que par les applications informatiques. Le format XML est le plus adapté aux développements d'applications WEB comme les EIAHs alors que le format MOF adresse davantage les développements en langage de programmation de base.
- Ouverture et extensibilité : les différents modèles CIM sont normalisés par le DMTF, et ouverts. Ils ont été conçus pour être étendus afin de satisfaire des objectifs précis tels que ceux visés par nos travaux.
- Réutilisation : CIM comprend notamment un schéma de base qui représente les fondements de toute modélisation. La spécification de modèles additionnels se réduit alors à l'identification des spécificités des entités mises en jeu dans le cadre de travail, et à la réutilisation/extension des schémas existants.
- Pérennité et support : la pérennité de CIM est assurée par les travaux réguliers de développement, de maintenance et de mise à jour des experts du DMTF. Aussi, le support est assuré par une communauté importante regroupant des acteurs majeurs de l'industrie informatique ainsi que des ingénieurs, chercheurs, et testeurs.

Dans le cadre de ce travail de recherche, nous avons proposé un ensemble de modèles génériques dédié aux télé-TPs indépendant du domaine d'apprentissage qu'il suffit d'étendre et de spécialiser pour couvrir les besoins d'un domaine particulier.

8.1.1.1. Modélisation des laboratoires

Nous avons proposé un modèle générique dédié aux laboratoires qui sont considérés comme des domaines administratifs sous l'autorité d'une institution. Les instances de ce modèle indiquent aux applications tierces (telles que les EIAHs) les différents laboratoires disponibles, la nature des dispositifs qu'ils mutualisent, ainsi que l'état de ses ressources. Ce modèle facilite ainsi le développement de systèmes de réservation de ressources en vue de

l'élaboration d'une expérience, de systèmes dédiés à la supervision et au suivi de la consommation des ressources mises en ligne par des laboratoires, etc.

8.1.1.2. Modélisation des expériences

Nous avons proposé un modèle pour décrire une expérience, fondé sur l'agrégation des dispositifs d'un ou plusieurs laboratoires. Le modèle d'une expérience est lié d'un côté au modèle des laboratoires d'où sont puisés les dispositifs nécessaires à la mise en œuvre de l'expérience, et d'un autre côté au modèle d'authentification pour identifier les apprenants et enseignants et leur offrir l'accès à une expérience dont les niveaux de privilèges correspondent au rôle des utilisateurs.

La puissance de cette modélisation réside dans la flexibilité offerte pour construire une expérience constituée de ressources disparates localisées dans des laboratoires distants et décentralisés. Aussi, la modélisation précise des ressources de laboratoires permet d'établir une vue globale et unifiée des ressources composant une expérience.

Le concepteur d'une expérience peut librement et facilement étendre les modèles génériques par dérivation et spécification des classes présentées dans le chapitre précédent, comme nous le démontrons dans la partie suivante qui s'intéresse au domaine de l'informatique.

8.1.1.3. Modélisation des Objets Pédagogiques Interactifs « Expérience »

La modélisation des objets pédagogiques « Expérience » diffère de la modélisation des expériences décrites ci-dessus car, même si ces deux modèles sont liés, le premier concerne le partage et la réutilisation d'expériences entre enseignants et applications pédagogiques, alors que le deuxième se focalise sur la gestion technique de ces expériences.

Un objet pédagogique interactif « Expérience » peut aisément être décrit par une fiche de métadonnées LOM, puisque sa modélisation comprend des informations classiques telles que sa description textuelle ou les objectifs d'apprentissage visés, mais également une description au format XML de l'expérience correspondant aux dispositifs nécessaires à la mise en place de l'activité de télé-TP. Le stockage, l'indexation et la réutilisation de ces objets pédagogiques au sein de viviers de connaissances existants sont donc assurés.

D'autre part, notre modélisation des OPIs « Expérience » précise les méthodes informatiques qui peuvent être invoquées sur ces entités et facilite ainsi leur intégration dans les plateformes d'exécution (voir ci-après).

8.1.1.4. Modélisation des traces d'activités

Les traces d'activités représentent un élément essentiel pour les applications d'analyse et d'évaluation des processus d'apprentissage. Profitant de la souplesse de notre approche, nous avons étendu un modèle générique de traces, conçu au sein de notre équipe de recherche, pour recueillir et exploiter les traces d'activités réalisées sur les expériences, en s'attachant à prendre en compte le caractère hautement interactif de ces dernières. Cette extension facilite la reconstitution du parcours d'un acteur humain sur une expérience non seulement en mode asynchrone (a posteriori), mais également en mode synchrone (en temps réel) pour supporter les activités pédagogiques de soutien et d'évaluation à l'apprenant.

8.1.2. Des interfaces de contrôle et de supervision uniformes

Les modèles présentés plus haut servent d'un côté à décrire les éléments du télé-TP, et d'un autre côté à diriger leurs contrôle et surveillance grâce aux éléments de l'architecture WBEM. Cette architecture normalisée par le DMTF offre une interface unique et uniforme pour la manipulation des objets CIM, et donc des dispositifs gérés au sein des laboratoires distants. Grâce à son interface de gestion, les applications de haut niveau (comme les EIAHs) sont libérées des détails spécifiques des éléments gérés : il suffit de disposer des modèles d'expériences pour pouvoir les exploiter sans en connaître les détails techniques de bas niveau.

En effet, grâce aux services de gestion des expériences en ligne présentés dans le chapitre 6, l'EIAH dispose d'un ensemble unique et minimum de fonctionnalités permettant la gestion de tout le cycle de vie des expériences ainsi que leur supervision. Cette interface de service, appropriée à n'importe quel domaine d'enseignement ou technologie de laboratoire en ligne, masque les détails techniques de ces dernières et simplifie ainsi le développement d'interfaces graphiques de haut niveau destinées aux utilisateurs finaux (staff technique, équipe pédagogique et apprenants), mais assure également la communication entre applications en vue de la réalisation de tâches complexes (à la manière de la composition de services web).

Enfin, cette interface permet de manipuler d'une manière uniforme et totalement transparente des dispositifs de laboratoires centralisés ou distribués sur plusieurs sites.

D'autre part, l'interface de gestion des laboratoires présente deux intérêts principaux : elle facilite le développement d'applications d'administration et de supervision au profit des administrateurs (tout en supportant les activités de conception et d'exploitation des expériences en ligne), et offre l'opportunité de construire une infrastructure de laboratoires virtuelle, composée d'une multitude de laboratoires physiques distribués à la manière des organisations virtuelles d'infrastructures de grilles de calcul. L'ensemble des laboratoires mutualisés apparaît comme un unique laboratoire satisfaisant un nombre plus important d'apprenants à la fois, et permettant d'accéder à des équipements inaccessibles à certains établissements.

8.2. Du point de vue pédagogique

Dans cette section nous mettons en avant les atouts apportés par notre approche d'un point de vue pédagogique. En particulier, nous nous intéressons aux différentes phases d'exploitation d'une expérience : intégration des activités de télé-TPs au milieu des autres activités du cursus d'apprentissage (pour faciliter l'acquisition des prérequis), conception d'une expérience par les enseignants, interaction des apprenants et tuteurs avec une expérience, et tutorat asynchrone.

8.2.1. Intégration et interaction de l'objet « Expérience » avec les EIAHs

Comme tout autre objet pédagogique, notre nouvel objet « Expérience » est *échangeable*, *réutilisable*, *indexable* et *autonome*. Par conséquent il devient exploitable dans tout système impliqué dans la formation à distance (outils auteurs de conception didactique/pédagogique, EIAH et viviers de connaissances).

Nous avons mentionné dans le chapitre 2 l'intérêt de séparer le concept d'expérience de celui du laboratoire, tant du point de vue théorique que technique. Cette distinction favorise l'exploitation des objets pédagogiques « Expérience » au sein des plateformes d'apprentissage puisque les laboratoires sont externalisés des EIAHs et accessibles de façon transparente pour opérationnaliser les activités de télé-TP.

8.2.2. Support à la conception d'expériences

Grâce au format ouvert des expériences fondé sur le langage XML, il est aisé de développer des outils auteurs dédiés à la conception visuelle des expériences ; dans le pire cas, il est possible d'utiliser des éditeurs XML génériques.

L'objet pédagogique « Expérience » peut être entièrement intégré dans le cycle de vie d'un scénario IMS-LD qui comprend :

- La modélisation par un outil auteur IMS-LD (comme Reload).
- L'exécution par un moniteur IMS-LD (comme CopperCore).
- L'analyse des traces a posteriori et la réingénierie des scénarios et objets d'apprentissage.

8.2.3. Support aux interactions avec les expériences et au tutorat synchrone

Le suivi et l'enregistrement automatique des traces d'activités et des identités associées rendent possible leur visualisation en temps réel à travers le service de gestion des traces. Ainsi, des outils de l'IHM ont la capacité de lister les utilisateurs impliqués dans une ou plusieurs expériences, et d'indiquer les différentes opérations qu'ils ont réalisées sur celle(s)-ci. Les tuteurs sont en mesure de facilement identifier les apprenants en difficulté, et d'exploiter les outils de communication synchrones et asynchrones nativement implantés dans les EIAHs actuels pour aider les apprenants. Aussi, l'IHM de contrôle met à la disposition des tuteurs, dans un objectif de démonstration pédagogique, des moyens d'intervention directe sur une expérience.

8.2.4. Des IHMs adaptées

Nous avons proposé une organisation structurelle d'une IHM dédiée à l'exploitation d'expériences en laboratoires distants qui intègre, dans une fenêtre unique, l'ensemble des fonctionnalités identifiées dans le chapitre 4 pour atteindre l'efficacité des activités de télé-TP.

D'autre part, notre modélisation abstraite des dispositifs de laboratoires et d'expériences offre l'opportunité de développer des IHMs personnalisées, voire contextualisées, selon différentes variantes :

- Des IHMs génériques dédiées à des classes de dispositifs hétérogènes modélisés par les mêmes classes CIM car ils sont pourvus des mêmes fonctionnalités.
- Des IHMs qui s'auto-adaptent dynamiquement au profil de l'apprenant en mettant en avant les éléments nécessaires à son apprentissage, et en cachant les éléments perturbateurs.
- Des IHMs qui agrègent en une seule interface les fonctionnalités de plusieurs dispositifs afin d'éviter aux apprenants de basculer entre différentes fenêtres de contrôle.

8.2.5. Support de l'apprentissage collaboratif et du tutorat asynchrone

Nos modèles et notre architecture favorisent l'implémentation de certains aspects de l'apprentissage collaboratif, en particulier en ce qui concerne l'*Awareness* des artefacts au travers du partage de l'IHM de télé-opération des expériences. Non seulement les utilisateurs sont en mesure de savoir « qui fait quoi », mais ils ont aussi l'opportunité d'agir ensemble sur une même ressource d'une expérience pour augmenter leur capacité à travailler en équipe.

Le tutorat asynchrone est pris en compte dans notre approche, comme le tutorat synchrone, par le suivi et le stockage des traces d'activités. En effet, à travers le service correspondant de la couche d'Intégration, des IHMs peuvent restituer l'historique des commandes opérées par chacun des apprenants, et permettre aux enseignants/tuteurs de concevoir des aides collectives et/ou personnalisées en fonction des erreurs constatées.

Partie III : Cas d'étude : l'enseignement des systèmes et réseaux informatiques

Chapitre 9. Un EIAH dédié aux travaux pratiques en ligne en informatique

9.1.	<u>Introduction au domaine d'enseignement visé</u>	190
9.1.1.	<u>Objectifs pédagogiques globaux d'un cursus informatique</u>	191
9.1.2.	<u>Les travaux pratiques en informatique</u>	192
9.1.3.	<u>Caractéristiques des laboratoires en ligne</u>	192
9.1.4.	<u>Un scénario type de télé-TP</u>	194
9.2.	<u>Méthodologie d'implémentation</u>	197
9.3.	<u>Les modèles CIM spécifiques aux expériences en informatique</u>	198
9.3.1.	<u>Modèle de gestion des ressources de laboratoires</u>	199
9.3.2.	<u>Modèle de gestion des expériences</u>	203
9.3.3.	<u>Modèle de l'OPI « Expérience »</u>	208
9.3.4.	<u>Modèle des traces d'activités</u>	209
9.3.5.	<u>Bilan</u>	212
9.4.	<u>Implémentation de la couche d'Apprentissage</u>	213
9.4.1.	<u>La plateforme d'apprentissage</u>	215
9.4.2.	<u>Le moteur d'exécution des expériences</u>	215
9.5.	<u>Implémentation de la couche d'Intégration</u>	218
9.5.1.	<u>Le serveur WBEM OpenPegasus</u>	220
9.5.2.	<u>Les services rendus à la couche d'Apprentissage</u>	220
9.6.	<u>Implémentation de la couche d'Expérimentation</u>	222
9.6.1.	<u>Le niveau inférieur</u>	223
9.6.2.	<u>Le niveau intermédiaire</u>	226
9.6.3.	<u>Le niveau supérieur : les Pilotes WBEM</u>	231
9.7.	<u>Scénario de coopération des différentes couches</u>	235
9.8.	<u>Synthèse</u>	236

Afin d'apporter des réponses aux verrous identifiés dans le chapitre 4, et notamment par rapport à ceux liés à l'apprentissage collaboratif et au tutorat en ligne, nous avons proposé dans la partie précédente une solution générique pour l'intégration des activités de télé-TP au sein des EIAHs existants. Dans cette dernière partie, nous montrons comment notre approche peut être utilisée dans le cadre de l'enseignement de l'informatique, et plus précisément dans celui des systèmes et des réseaux. En effet, nous partageons l'avis de P. Leroux à propos de l'influence du domaine d'application sur le développement de la recherche en EIAH (Leroux, 2002) : « *La logique scientifique de la recherche en informatique est de contribuer le plus possible à la proposition de modèles, de spécifications et de systèmes génériques ou réutilisables. Néanmoins, pour des questions de coûts de développement et d'expérimentation (en terme de temps, d'investissement dans un domaine, du nombre de chercheurs susceptibles d'être mobilisés sur un projet, etc.), il est actuellement difficile de multiplier les contextes d'usage. C'est pourquoi souvent les équipes de recherche en EIAH ont un ou des domaines d'étude ou d'application privilégiés pour leurs travaux ; par exemple, nous pouvons citer la géométrie, l'algèbre et la programmation à Grenoble, les mathématiques et la physique à Lyon, l'algèbre, le français langue étrangère et la RP⁴³ au Mans* ».

Ce chapitre détaille les travaux de développement réalisés pour implémenter l'ensemble des couches de notre architecture dans le cadre de l'apprentissage de l'informatique, alors que le chapitre suivant se concentre sur les outils et IHMs de la couche d'Apprentissage pour le support des activités pédagogiques. Cette partie est conclue par les résultats d'une expérimentation menée auprès d'un public d'ingénieurs.

9.1. Introduction au domaine d'enseignement visé

L'informatique est une discipline scientifique définie par le JTFCC⁴⁴ comme « *toute activité nécessitant, bénéficiant, ou créant un système informatique*⁴⁵. Elle inclut la conception et la

⁴³ Robotique Pédagogique

⁴⁴ The Joint Task Force for Computing Curricula, constitué conjointement par l'IEEE et l'ACM dont l'objectif est de produire des standards non prescriptifs sur la façon de construire des cursus en informatique.

⁴⁵ « Système Informatique » signifie « Computer System » en anglais.

construction de systèmes matériels ou logiciels pour un large éventail de besoins : traitement, structuration et gestion de différents types d'informations, toute étude scientifique utilisant des ordinateurs, rendre des systèmes informatiques intelligents, création et utilisation de médias de communication et de divertissement, recherche et recueil d'informations pour tout usage particulier, etc. »⁴⁶ (JTFCC, 2005). Aussi, les auteurs du rapport (CBG, 2000) considèrent l'informatique comme « *la discipline qui a trait à la compréhension, la conception et l'exploitation de systèmes informatiques matériels et logiciels* ».

9.1.1. Objectifs pédagogiques globaux d'un cursus informatique

L'informatique est caractérisée par un important corps de connaissances dont une description détaillée figure dans divers rapports de groupes de travail chargés de proposer des recommandations sur les contenus des cursus en informatique (JTFC, 2005 ; CBG, 2000 ; ABET, 2009b). Le rapport (CBG, 2000) propose une classification fondée sur deux dimensions : la dimension Matériel-Logiciel et la dimension Théorie-Pratique. D'une manière générale, les étudiants en informatique sont invités à développer un large éventail d'habiletés et de compétences que le groupe de travail (CBG, 2000) divise en trois grandes catégories :

- Les habiletés et compétences cognitives regroupent l'ensemble des connaissances théoriques, des compétences de modélisation de systèmes informatiques, d'identification des besoins et des contraintes liés à un problème informatique donné, l'évaluation et la critique d'un système informatique existant, ainsi que l'utilisation et le déploiement de méthodes et outils appropriés.
- Les compétences pratiques regroupent toutes les compétences nécessaires à la spécification, la modélisation et la construction de systèmes informatiques, la capacité d'identifier et de déployer d'une manière efficace les outils pour la résolution d'un problème informatique, l'aptitude à exploiter efficacement le matériel informatique compte tenu de ses propriétés physiques et logiques, l'aptitude à évaluer les systèmes informatiques en terme de qualité générale, la capacité à reconnaître les risques éventuels ou les aspects de sécurité liés aux installations informatiques, et enfin la capacité à travailler comme membre d'un groupe dans le cadre d'un projet.

⁴⁶ Traduction libre

- Enfin, les compétences générales regroupent les compétences liées au traitement et à l'extraction efficaces de l'information, à l'utilisation pertinente des outils de la technologie de l'information et de la communication, à la capacité de gérer son apprentissage et à la formation continue.

Cette classification montre que les objectifs pédagogiques présentés sous forme de compétences attendues correspondent à la classification des objectifs pédagogiques présentés dans le chapitre 2, mais appliquée à notre domaine d'enseignement. En particulier, nous notons l'importance accordée aux compétences de modélisation qui se situent entre les compétences cognitives et pratiques. Une importance soutenue est également accordée aux compétences sociales, de communication et de documentation, liée à la pression du monde professionnel sur les établissements de formation pour former des étudiants possédant des compétences non techniques (JTFCC, 2004).

9.1.2. Les travaux pratiques en informatique

La pratique occupe une place majeure tout au long du cursus car contrairement à d'autres disciplines, les diplômés en informatique peuvent rentrer dans le monde professionnel dès la fin du premier cycle de formation (JTFCC, 2004). Il est primordial pour les étudiants de cette discipline d'avoir la possibilité « *d'observer, d'explorer et de manipuler les caractéristiques et le comportement des périphériques, des systèmes et des processus* » à travers « (1) *la modélisation, la conception et les tests*, (2) *la documentation du logiciel et du matériel*, et (3) *la modélisation des expériences pour recueillir des données, analyser et interpréter ces données et éventuellement les utiliser pour améliorer la modélisation* » (JTFCC, 2004).

9.1.3. Caractéristiques des laboratoires en ligne

Les composants d'un laboratoire en informatique peuvent être mis à disposition des utilisateurs distants de la même façon que les autres types de laboratoires selon l'architecture générique présentée dans la section 2.4.3.1 (voir la Figure 2-5). La Figure 9-1 illustre l'architecture d'un laboratoire informatique en ligne où chaque ressource du laboratoire est reliée au serveur de médiatisation du laboratoire via deux connexions :

- Une ou plusieurs interfaces de communication (de type Console, Ethernet, Token-Ring, Wifi, etc.) accessibles aux apprenants et dédiées à la télé-opération des activités de télé-TP.
- Une interface de communication (de type réseau, série, USB, etc.) dédiée à l'administration et inaccessible aux apprenants.

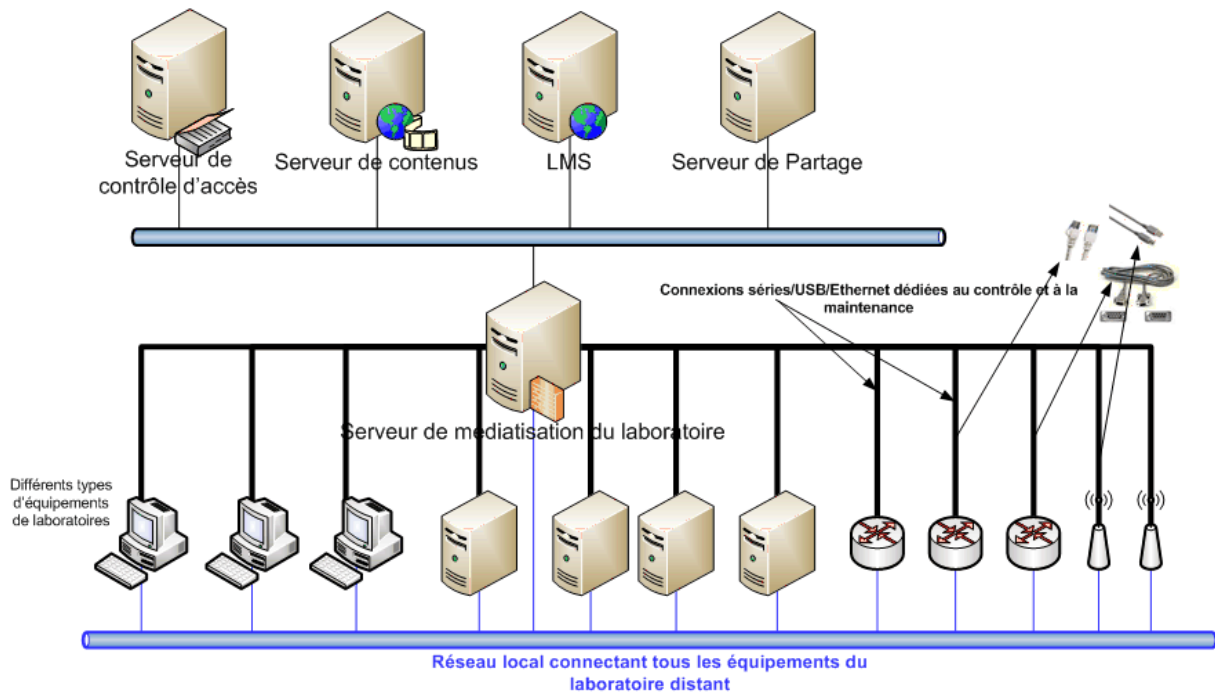


Figure 9-1. Architecture d'un laboratoire en informatique en ligne.

La télé-opération d'un laboratoire informatique dans le cadre d'un télé-TP s'effectue à travers les outils utilisés pour télé-opérer des systèmes informatiques en laboratoire classique ou en production. Le réalisme de ces moyens est une condition nécessaire à l'efficacité pédagogique en favorisant l'acquisition efficace d'habiletés techniques et pratiques. Il existe généralement deux types d'outils pour télé-opérer des systèmes informatiques :

- Les interfaces de type ligne de commande (CLI) qui utilisent les protocoles TELNET, SSH ou Virtual Terminal.
- Les interfaces graphiques comme le bureau Windows à distance, l'interface VNC⁴⁷ ou les interfaces X11 sous Unix/Linux utilisant des protocoles de même nom.

⁴⁷ Virtual Network Computing

Même si les outils et protocoles mentionnés ci-dessus peuvent satisfaire des objectifs de supervision à travers l'interrogation « manuelle » des différents composants de l'expérience, des moyens existants et élaborés pour superviser des systèmes informatiques en production sont généralement préconisés. Ces outils sont fondés essentiellement sur les protocoles SNMP, CMIP et CIM/WBEM. Comme exemple citons les logiciels propriétaires HP OpenView, Cisco Works et IBM Tivoli, ainsi que les logiciels libres MRTG, Nagios ou CACTI.

Notons que l'architecture de la Figure 9-1 s'applique également aux laboratoires émulés fondés sur les technologies de virtualisation où les interfaces de communication physiques sont remplacées par des interfaces virtuelles.

9.1.4. Un scénario type de télé-TP

Afin d'illustrer le fonctionnement de notre système tout au long de cette partie, nous proposons ici une activité de travaux pratiques typique dans le domaine des réseaux : il s'agit d'un télé-TP sur les principes de routage statique dans les réseaux IP.


Connecté sous le nom « amine bouabid » (Déconnexion)

Accueil ► Mes cours ► Télé-TP-Sys-Net

Navigation

- Accueil
- Ma page
- Pages du site
- Mon profil
- Mes cours
 - Télé-TP-Sys-Net
 - Participants
 - Rapports
 - Généralités
 - Forum des nouvelles
 - Forum du Télé-TP
 - Salon de Chat Général
 - Section 1
 - Section 2

Réglages

- Administration du cours
 - Activer le mode édition
 - Paramètres
 - Utilisateurs
 - Filtres
 - Notes
 - Objectifs
 - Sauvegarde
 - Restauration
 - Importation
 - Réinitialisation
 - Banque de questions
- Prendre le rôle...
- Réglages de mon profil

Aperçu des sections

Introduction à l'activité

Le routage est le processus par lequel le trafic réseau est dirigé vers sa destination finale par l'utilisation de la couche 3 du modèle OSI. Dans ce télé-TP, nous allons examiner comment cela fonctionne, et mettre en œuvre le routage statique (configuré manuellement)

Forum des nouvelles

Objectifs Spécifiques du Télé-TP

Comprendre les principes généraux relatifs aux réseaux (modèle OSI)
Comprendre le protocole TCP/IP et le modèle de réseau TCP/IP.
Définir et mettre en place un adressage IP et un routage IP.
Maîtriser les outils réseaux (ifconfig, netstat, ping, traceroute, etc..)
Maîtriser les outils de communications (telnet, rlogin, ssh, ftp, etc..).
Identifier et résoudre les dysfonctionnements d'un réseau.
Comprendre le principe du routage,
Configurer un serveur Linux en tant que passerelle/routeur.

Outils de Communications Générales (Forum et Chat)

Forum du Télé-TP
Salon de Chat Général

1

Pré-Lab: Quelques Rappels Théoriques

- Cours TCP/IP
- Adressage IP

Accès à l'Expérience en Ligne

Consignes Générales:

- Consulter les supports de cours fournis
- Consulter et compléter le document donné comme ressource
- S'entendre sur une méthode de travail d'équipe pour configurer les composants de l'expérience en exploitant les outils de contrôle fournis: accès simple, accès en lecture et accès partagé

- Description de l'Expérience
- Manuel des commandes Linux: ifconfig et route
- Document de Travail
- Manuel du Système
- Interface d'Accès à l'Expérience

2

Post-Lab: Dépôt des Comptes rendus et Passage de tests

- Dépôt du Compte Rendu
- Enquête d'Evaluation Pour Apprenants
- Enquête D'évaluation

3

Activer le mode édition

Utilisateurs en ligne
(5 dernières minutes)

- amine bouabid

Utilisateurs en ligne
(5 dernières minutes)

- amine bouabid

Utilisateurs en ligne
(5 dernières minutes)

- amine bouabid

Figure 9-2. Scénario d'un télé-TP traitant des réseaux informatiques.

La Figure 9-2 ci-dessus présente l'activité de télé-TP telle qu'elle apparaît dans l'interface de la plateforme d'enseignement, et se compose d'un certain nombre d'outils et de ressources

généralement délivrés dans ce genre d'activité : une introduction générale, des objectifs spécifiques, des outils de communication et de collaboration, des rappels théoriques (la phase *Pré-Lab*), une expérience en laboratoire, et un test suivi d'un compte-rendu (la phase *Post-Lab*).

L'expérience en ligne constitue le cœur de notre contribution puisqu'elle représente la ressource caractéristique distinguant un télé-TP des autres types d'activités d'apprentissage. L'expérience en laboratoire, dans notre scénario d'apprentissage, se compose des éléments suivants :

- La description de l'activité explicite l'architecture réseau de l'expérience ainsi que les tâches à réaliser par les apprenants. Dans notre cas, l'architecture de l'expérience illustrée par la Figure 9-3 correspond à une interconnexion de deux réseaux locaux (*LAN*) composés chacun de machines connectées à deux commutateurs, et dont l'interconnexion est assurée par un routeur ; l'un des LAN est protégé par un pare-feu.
- Un manuel des commandes Linux *ifconfig* et *route*.
- Le manuel détaillé de l'interface d'expérimentation (voir Annexe F).
- L'énoncé détaillé de l'activité sous la forme d'un document à remplir tout au long de l'activité, et à déposer dans l'espace *Post-Lab* lors de l'accomplissement de l'activité (voir Annexe F, section 4).
- La fenêtre d'accès à l'expérience, sous la forme d'une IHM de télé-opération et de supervision.

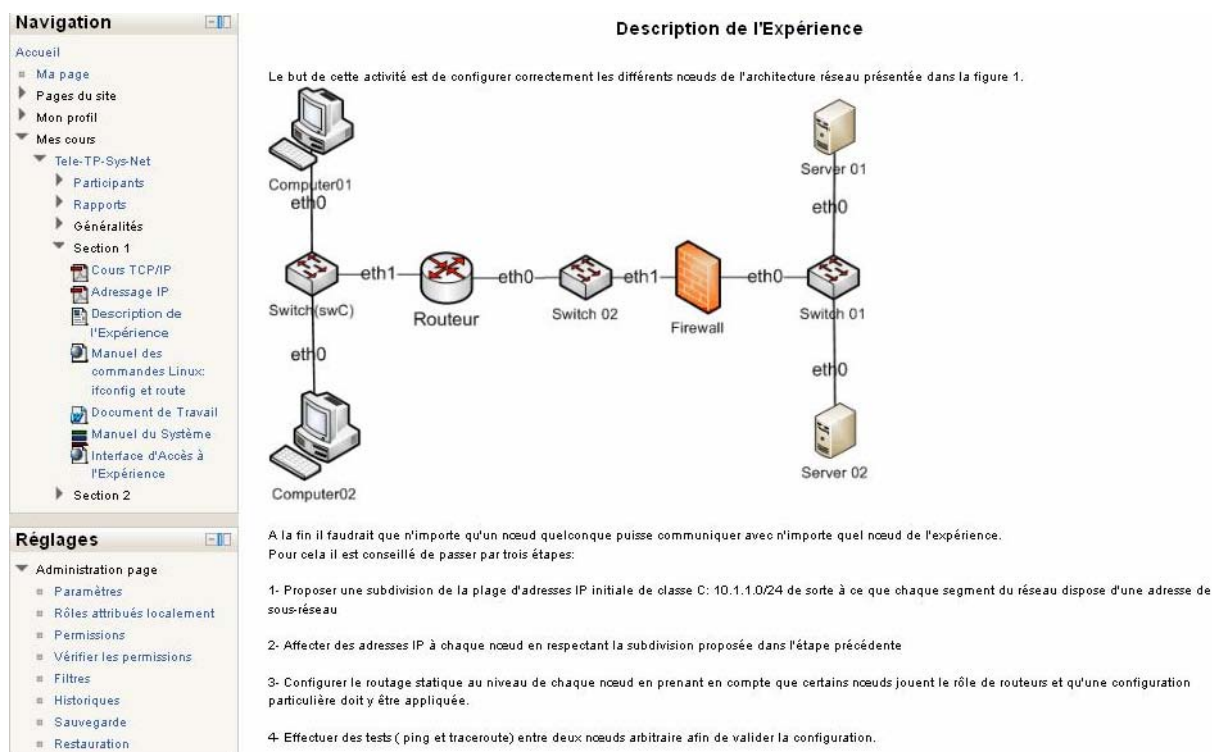


Figure 9-3. Description de l'expérience dans le cadre de notre étude.

Dans la suite de ce chapitre nous nous focalisons sur l'implémentation de l'expérience en laboratoire, puisque les autres activités sont nativement fournies par la plateforme d'apprentissage.

9.2. Méthodologie d'implémentation

La méthodologie que nous avons adoptée pour implémenter l'expérience en laboratoire est une méthodologie modulaire qui consiste à développer les modules correspondants aux composants de l'architecture présentée dans le chapitre 5. D'une manière générale, notre approche est fondée sur la philosophie du système UNIX (Salus, 1994) qui consiste à sélectionner, et au besoin développer, des outils ou des modules logiciels spécialisés dans des fonctions précises et de les combiner pour construire un système plus complexe. Pour atteindre ces objectifs, nous avons suivi les règles générales suivantes (Metcalf, 2011) :

- Identifier les besoins, les contraintes et les outils existants, et les positionner dans notre architecture générale.
- Identifier les composants à concevoir pour construire notre système.
- Pour chacun de ces composants, effectuer une recherche et une sélection d'outils implémentant tout ou partie des fonctionnalités du composant manquant. Lorsque

plusieurs outils existent, faire un choix sur la base d'un rapport qualité/fonctionnalités fondé sur l'utilité, l'extensibilité et le respect des normes, la scriptabilité, la maturité, etc.

- Lorsque certaines fonctionnalités ne sont pas implémentées, procéder au développement de celles-ci soit par la modification ou l'extension de code existant, soit par un développement spécifique du composant manquant.

Avant de rentrer dans les détails d'implémentation de chacune des trois couches, nous présentons ci-après des modèles qui étendent ceux présentés dans le chapitre 7, et qui assurent la prise en charge des expériences en ligne dans le domaine de l'informatique.

9.3. Les modèles CIM spécifiques aux expériences en informatique

Nous présentons dans cette section les détails des modèles CIM respectivement dédiés à la gestion des ressources de laboratoires, à la gestion des expériences et à la gestion des traces d'activités, et qui sont spécifiques à l'informatique. Ces modèles constituent la « colle » entre les trois couches de notre architecture car ils sont associés d'un côté aux objets pédagogiques interactifs de type « Expérience » manipulés par la plateforme d'apprentissage, et d'un autre côté aux dispositifs de laboratoire de la couche d'Expérimentation, comme l'illustre la Figure 9-4.

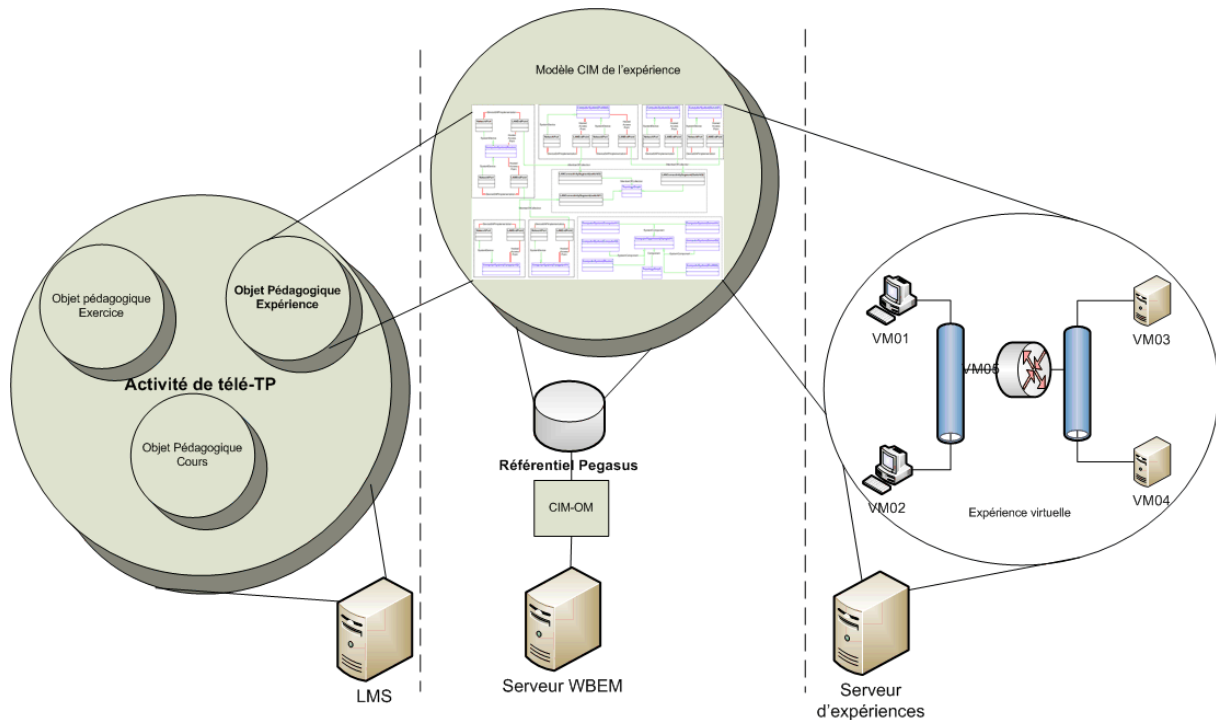


Figure 9-4. Correspondance entre objets pédagogiques, objets CIM et dispositifs de laboratoire.

9.3.1. Modèle de gestion des ressources de laboratoires

Le modèle de gestion des ressources que nous proposons ici est une spécialisation du modèle générique présenté dans la section 7.2.1. Afin de gérer efficacement les ressources informatiques partagées, nous nous sommes inspirés des domaines de gestion des ressources des grilles de calcul et des nuages informatiques (*Cloud Computing*) qui comptabilisent les ressources informatiques élémentaires telles que les processeurs, la mémoire vive, l'espace disque, les différents périphériques, etc. De plus, afin d'optimiser et de simplifier la gestion des ressources informatiques, nous avons opté pour une solution intuitive qui consiste à fixer dès le départ le nombre et la taille des ressources virtuelles en fonction des ressources physiques disponibles. Nous avons également adopté une allocation par défaut des ressources aux nouvelles machines virtuelles, sauf si un besoin précis est explicitement exprimé dans le modèle de l'expérience.

Nous avons identifié diverses classes CIM natives qui modélisent les ressources mentionnées dans le paragraphe ci-dessus :

- *Processor* modélise les processeurs (DMTF, 2003d ; 2003e).

- *StorageExtent* (DMTF, 2000) représente n'importe quel périphérique de stockage, alors que d'autres classes dérivées spécifient les périphériques les plus communs tels que la mémoire vive (*Memory*), les disques durs (*LogicalDisk*), les partitions de disque (*Partition*), les volumes de stockage (*Volume*) ou les lecteurs CD/DVD.
- *NetworkPort* (DMTF, 2003b; 2006b; 2006c) modélise n'importe quel adaptateur réseau.
- *ComputerSystem* (DMTF, 2006a) modélise n'importe quel système informatique : ordinateur personnel, serveur, routeur, pare-feu, commutateur, etc.

La Figure 9-5 illustre notre extension du modèle générique de gestion des ressources présenté par la Figure 7-5, appliquée aux ressources informatiques ; pour en alléger la lisibilité, les classes déjà illustrées par le modèle générique ont été éliminées. Ce modèle spécifique comprend quatre ressources fondamentales, respectivement les périphériques et autres ressources matérielles (classe *LogicalDevice*), les services (classe *Service*), les logiciels (classe *SoftwareIdentity*) et les systèmes (classe *System*), à partir desquelles nous avons intégré d'autres classes afin d'exprimer des propriétés matérielles et logicielles essentielles. Il s'agit des classes qui modélisent:

- Les interfaces et adaptateurs réseaux décrivant les technologies les plus utilisées : Ethernet (classe *EthernetPort*), TokenRing (classe *TokenRingPort*), sans fil (classe *WirelessPort*) et *infini band* (classe *IBport*).
- Les processeurs et microprocesseurs modélisés par la classe *Processor*.
- Les différents types de mémoires et espaces de stockage à travers les classes décrivant la mémoire vive, les disques durs, les partitions et les volumes.
- Tout type de système informatique : machines physiques, terminaux, serveurs, routeurs, commutateurs, machines virtuelles, systèmes embarqués, etc. (représentés par la classe *ComputerSystem*).

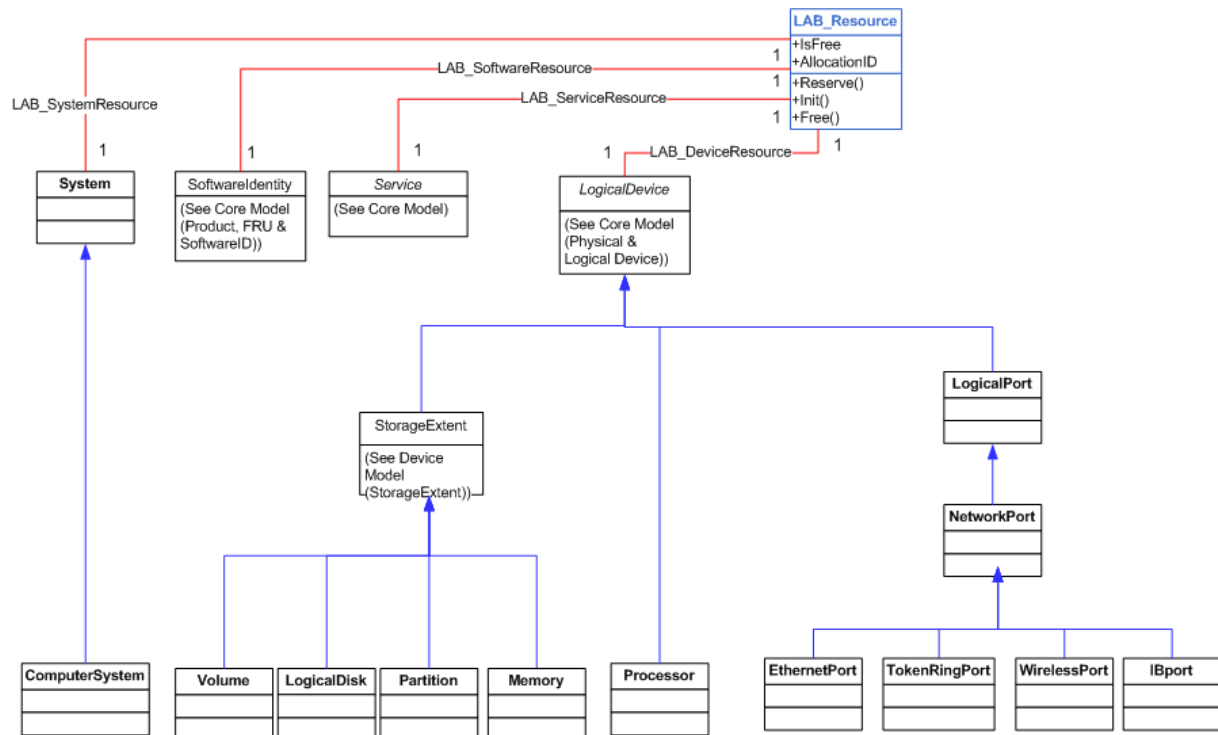


Figure 9-5. Modèle de gestion des ressources d'un laboratoire informatique.

Dans notre scénario type, la machine *Computer01* est constituée d'une ressource de type CPU (*CPU-02*), d'une ressource de type mémoire (*RAM-02*) et d'une ressource de type disque (*DISK-02*), intégrées dans une ressource de type système informatique (*CS-02*) ; la Figure 9-6 illustre le diagramme d'instances correspondant, les autres ressources étant affectées à d'autres équipements de l'expérience.

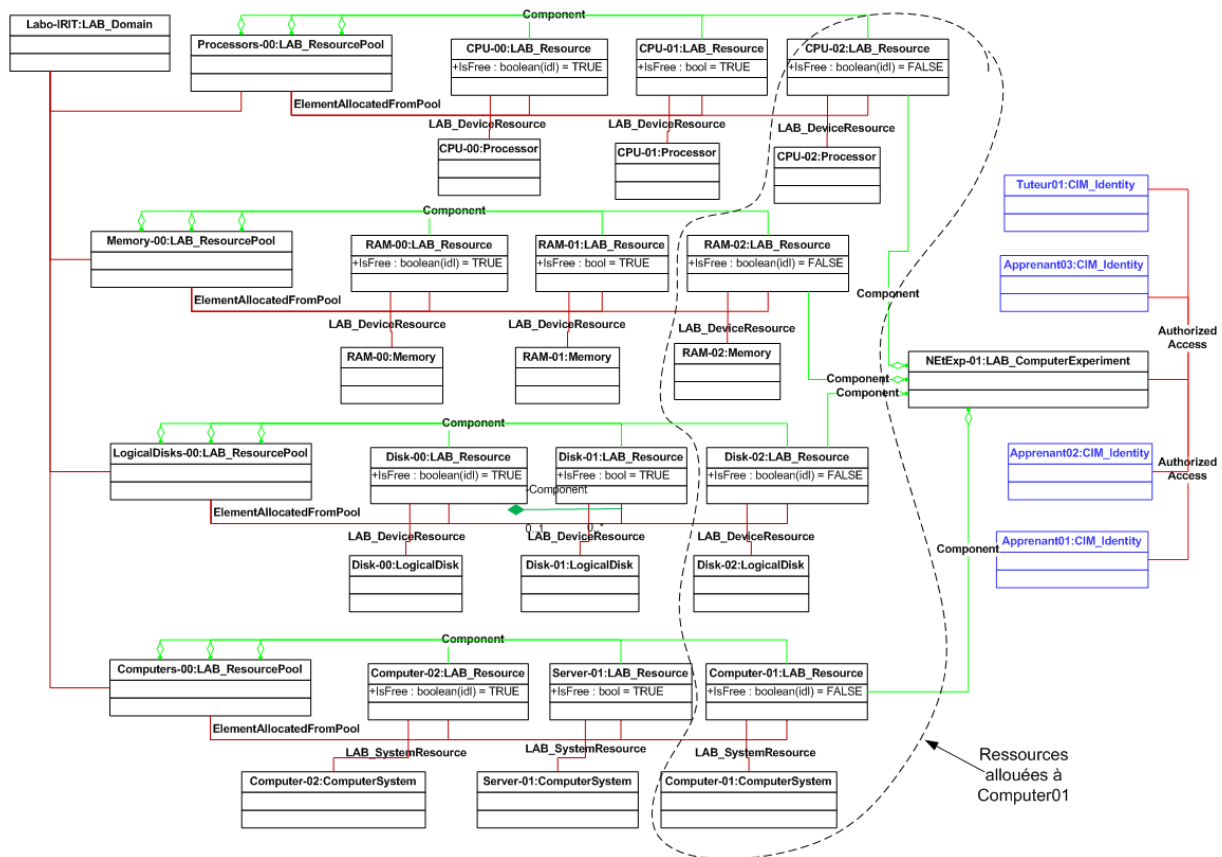


Figure 9-6. Diagramme d'instances CIM de ressources de laboratoire.

Ce modèle spécifique à la gestion des ressources informatiques ne fait aucune supposition sur la nature physique ou virtuelle de celles-ci : du point de vue pédagogique, les deux scénarios présentent le même diagramme d'instances de ressources. D'autre part, il est important de rappeler que ce modèle est assez souple pour être étendu à d'autres classes décrivant d'autres propriétés physiques ou logiques ou d'autres périphériques, grâce notamment aux propriétés d'héritage et de polymorphisme dont bénéficie le méta-modèle CIM. Aussi, les classes d'association que nous avons introduites dans le modèle générique (*LAB_SystemResource*, *LAB_SoftwareResource*, *LAB_ServiceResource* et *LAB_DeviceResource*) suffisent à établir des liens entre de nouvelles ressources et les classes *LAB_ResourcePool* et *LAB_Resource* dédiées à la gestion de celles-ci.

Remote et l'attribut *RemoteHost* contenant son nom ou son adresse IP. Le Tableau 9-1 résume les valeurs de ces attributs dans les cas d'expériences locales et distribuées.

Tableau 9-1. Combinaison des valeurs des attributs dédiés à la gestion d'expériences locales ou distribuées.

Instance CIM	Attribut	Valeur de l'attribut pour une expérience locale	Valeur de l'attribut pour une expérience distribuée
<i>LAB_ComputerExperiment</i>	<i>Distributed</i>	<i>False</i>	<i>True</i>
<i>LAB_TopologyGraph</i>	<i>Distributed</i>	<i>False</i>	<i>True</i>
<i>LAB_ComputerSystem</i>	<i>Remote</i>	<i>False</i>	<i>True</i>
	<i>RemoteHost</i>	<i>NULL</i>	Nom de domaine ou adresse IP de la machine distante

9.3.2.2. Le modèle détaillé d'une expérience réseau

Le modèle détaillé que nous proposons pour les expériences réseau est illustré par la Figure 9-8 : il comprend notre modèle de gestion des ressources présenté plus haut, et intègre un ensemble de modèles natifs du DMTF qui modélisent quatre types d'éléments :

- Les composants matériels (DMTF, 2006a) : processeurs, mémoires, disques et interfaces réseaux.
- Les composants logiciels (DMTF, 2006a) : le système d'exploitation d'une machine ainsi que le système de fichier associé sont modélisés par les classes natives *OperatingSystem* et *FileSystem*. Les autres logiciels héritent de la classe native *SoftwareIdentity*.
- Les composants réseau d'une machine (DMTF, 2003b ; DMTF, 2006b ; DMTF, 2006c) : propriétés physiques et logiques des différentes couches du modèle TCP/IP.
- La topologie du réseau (DMTF, 2011) : l'architecture du réseau à tous les niveaux du modèle TCP/IP.

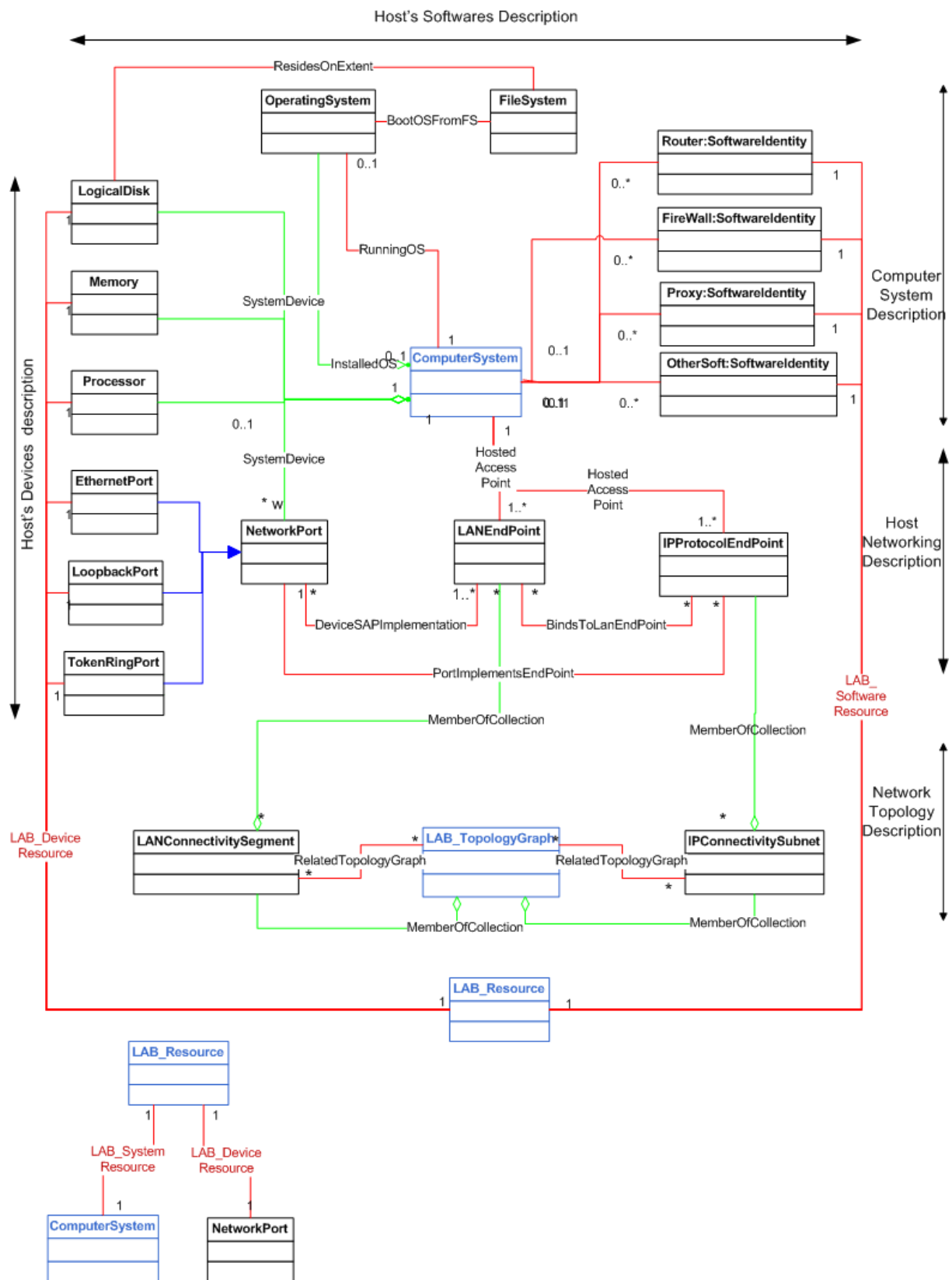


Figure 9-8. Modèle détaillé d'une expérience réseau.

Notre contribution réside ici essentiellement dans l'élaboration d'un noyau intégrant et associant des classes et associations natives avec nos classes de gestion des laboratoires et des expériences. Ceci n'exclut pas la possibilité de proposer d'autres modèles qui expriment d'autres points de vue de gestion. En effet, le modèle de gestion des ressources est complètement indépendant du modèle de supervision des expériences : le premier est destiné à supporter les processus administratifs d'identification, de réservation, d'allocation et de libération des ressources, alors que le second supporte la supervision et l'exploitation des expériences pour atteindre des objectifs pédagogiques.

La Figure 9-9 illustre le diagramme d'instances CIM-UML global de notre expérience type ; toutefois, pour des raisons de lisibilité, nous n'avons illustré que les instances des nœuds et de leurs ports réseaux, ainsi que les instances relatives à la topologie. En bas à droite de la figure, une instance de la classe *LAB_ComputerExperiment* nommée *NetExp01*, qui modélise l'expérience dans sa globalité, est reliée aux instances des composants de l'architecture réseau illustrée par la Figure 9-3 (il s'agit des nœuds *Computer01*, *Computer02*, *Server01*, *Server02*, *Routeur* et *Firewall*). Au centre de la figure, l'instance de la topologie (*TP01*) est reliée aux trois instances de la classe *LANConnectivitySegment* qui représentent les trois segments réseaux de notre expérience ; ces dernières modélisent en fait les trois commutateurs virtuels de notre expérience (*Switch01*, *Switch02* et *Switch03*). Un segment réseau est considéré comme une collection de ports rattachés aux nœuds de l'architecture réseau. Dans notre scénario type, pour modéliser les liens du réseau, des associations de type *MemberOfCollection* relient chaque instance de la classe *LANEndPoint* (décrivant une extrémité réseau au niveau d'un nœud) à l'instance d'un commutateur. Grâce à différentes associations masquées pour des raisons de lisibilité, il est possible de retrouver l'instance du port physique (*NetworkPort*) et l'instance de l'extrémité IP associée (*IPProtocolEndPoint*) qui encapsule ses paramètres IP (adresse IP, masque de sous-réseau, etc.). Enfin, l'association *HostedAccessPoint* permet d'identifier l'instance du système (*ComputerSystem*) qui héberge tous les éléments physiques et logiques précédents.

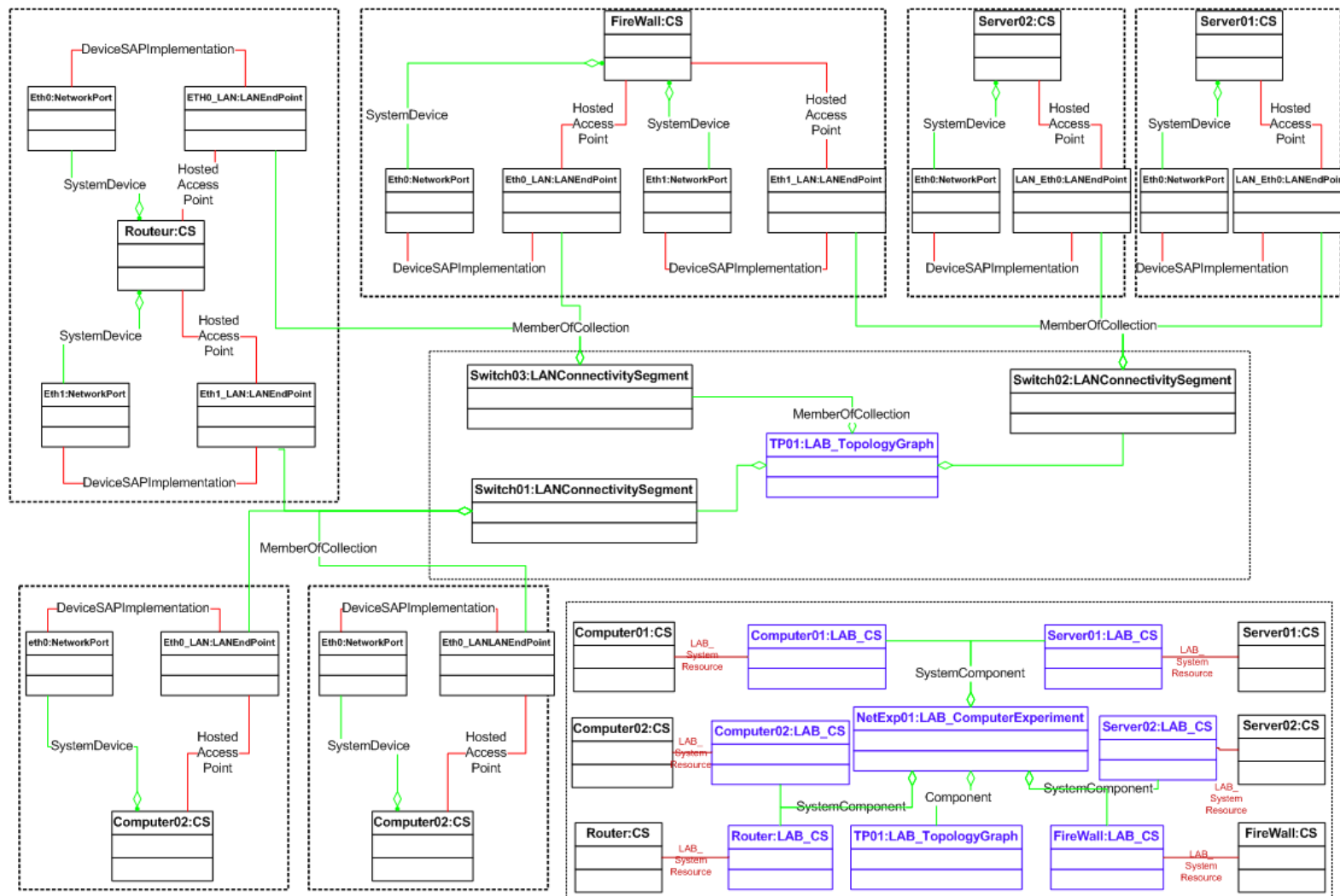


Figure 9-9: Diagramme d'instances CIM-UML de notre expérience type.

9.3.3. Modèle de l'OPI « Expérience »

Nous avons présenté dans la section 7.3 la classe *TEL_Experiment* qui modélise un Objet Pédagogique Interactif « Expérience ». Rappelons que l'intérêt de ce modèle est de produire des objets pédagogiques exploitables par les EIAH au niveau de la couche d'Apprentissage, mais également de spécifier, à travers l'attribut *Template*, une description a priori du modèle de gestion de l'expérience compréhensible par la couche d'Intégration. Pour notre domaine d'enseignement, nous proposons deux nouvelles classes associées entre elles afin de représenter notre OPI « Expérience » spécifique à l'informatique. Il s'agit respectivement de :

- *TEL_ComputerSystem* spécifie a priori un nœud dans une expérience.
- *TEL_ComputerExperiment* définit a priori l'expérience globale.

La relation entre ces deux classes est présentée dans la Figure 9-10 qui illustre également la relation de ce modèle avec le modèle de gestion des expériences présenté dans la section précédente.

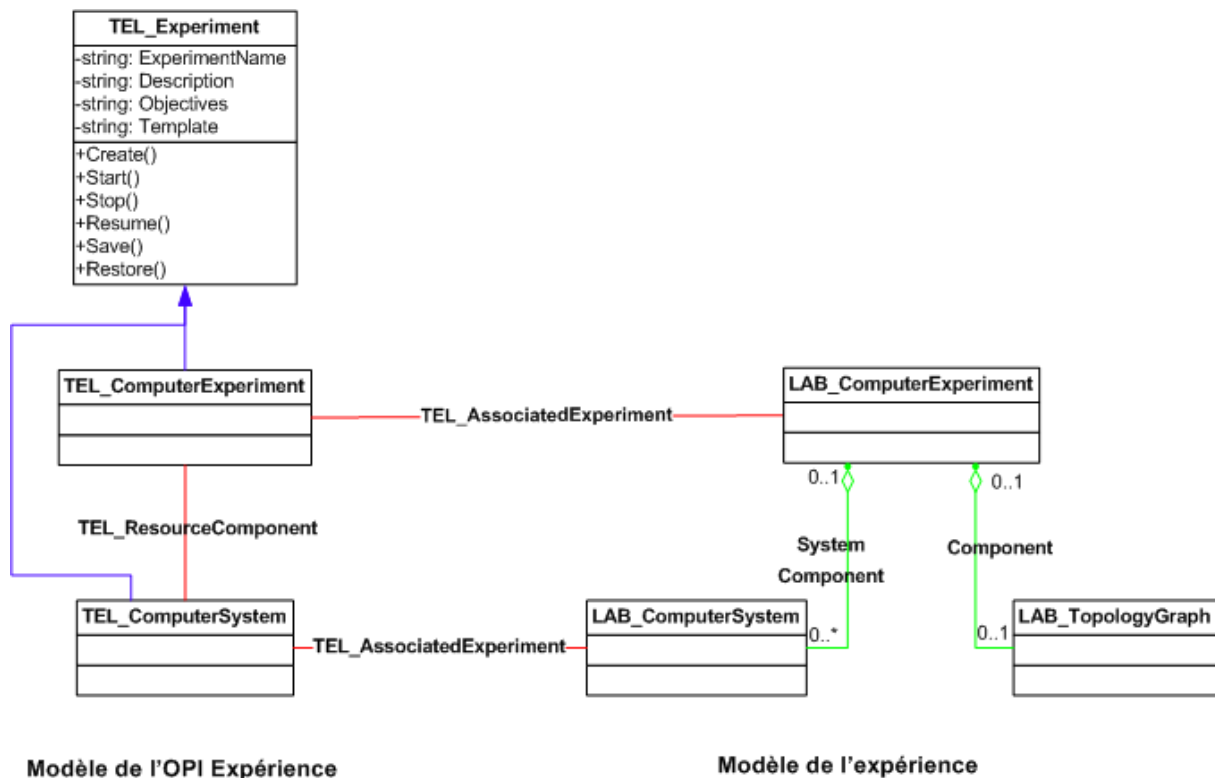


Figure 9-10 : Modèle de l'OPI « Expérience » en informatique, et sa relation avec le modèle de gestion des expériences.

Pour illustrer ce modèle, nous donnons dans la Figure 9-11 le diagramme d'instances des OPI « Expérience » correspondant à notre expérience type : au milieu sont représentées les instances constituant l'OPI « Expérience » (composé essentiellement d'objets de types *TEL_ComputerExperiment* et *TEL_ComputerSystem*), à gauche le contenu de leur attribut *Template*, et à droite les instances de l'expérience opérationnelle (composé essentiellement d'objets de types *LAB_ComputerExperiment* et *LAB_ComputerSystem*).

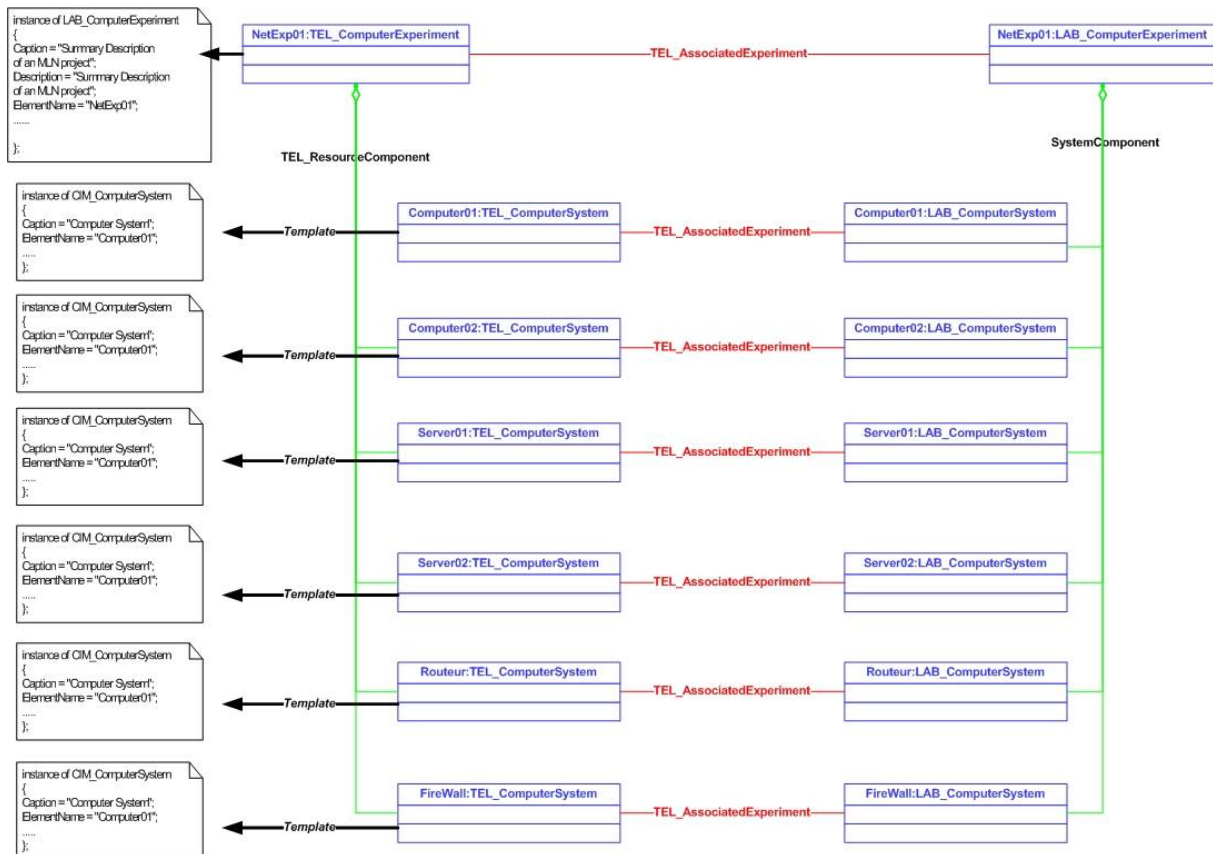


Figure 9-11 : Diagramme d'instances de l'OPI « Expérience » correspondant au scénario type.

9.3.4. Modèle des traces d'activités

Un modèle de traces d'activités dépend du type d'interface utilisé pour l'interaction entre l'utilisateur et le dispositif de l'expérience. Dans le contexte de l'informatique, l'interaction avec une machine s'effectue via deux modalités : la ligne de commande et les différentes interfaces graphiques existantes. La solution la plus complète consiste bien sûr à proposer des interfaces graphiques évoluées, mais vu la complexité et l'ampleur des tâches de développement associées, nos travaux se limitent, dans un premier temps, à l'interaction avec

un Terminal de type texte orienté caractères. En effet, ce moyen de commande est le plus ancien mais il demeure très puissant et très répandu parmi les utilisateurs et les développeurs. D'autre part, d'un point de vue pédagogique, le Terminal reste un passage obligé pour les étudiants en informatique qui doivent être initiés à tous les moyens de commande et d'exploitation d'un système informatique.

Alors à partir du modèle générique des traces proposé dans la section 7.4.2, nous avons créé la classe *TEL_CommandExperimentActivity* présentée par la Figure 9-12 et dérivée de la classe *TEL_ExperimentActivity*. Elle est dédiée aux activités de type ligne de commande et se caractérise par les attributs suivants :

- *Command* correspond aux caractères entrés par l'utilisateur ou au(x) message(s) retourné(s) par le système.
- *SeqId* indique le numéro de séquence de la commande dans le flux des commandes d'une session.
- *Initial* et *Final*, tous deux de type booléen, sont respectivement positionnés à *True* uniquement pour les première et dernière commandes.
- *ExpId*, *HostId* et *SessId* identifient respectivement l'expérience, la machine et la session dans lesquelles la commande est exécutée.
- *UserId* spécifie l'utilisateur qui a tapé la commande ou reçu le message.

Aussi, pour modéliser les sessions de travail de type Terminal, nous avons introduit la classe *TEL_TTYSession* qui hérite de la classe *TEL_SessionExperimentActivity* présentée dans la section 7.4.2. Il est alors possible, à partir de cette classe, de retrouver la première instance des commandes émises lors d'une session et de parcourir les autres jusqu'au dernier enregistrement, et ainsi reconstituer le déroulement d'une session terminée (a posteriori) ou consulter une session active (en temps réel). Notons que la classe *TEL_GraphicalSession* est dédiée à la modélisation des sessions graphiques sur n'importe quel système offrant ce genre d'interfaces de commandes.

sur ce(s) nœud(s). Précisons enfin que nous n'avons rien rajouté par rapport à la modélisation/gestion des utilisateurs et EIAHs puisque le modèle global présenté dans les sections 7.3 et 7.4 demeure suffisant.

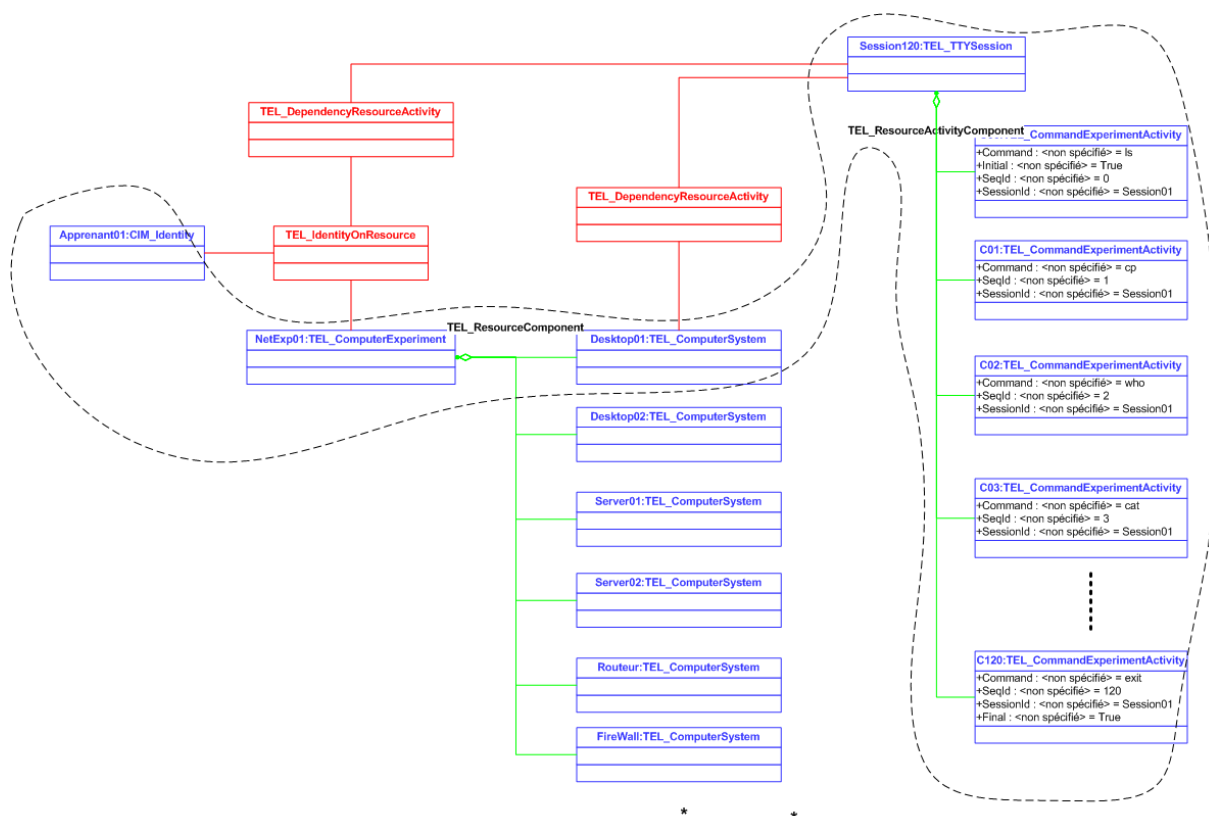


Figure 9-13. Diagramme d'instances d'une session sur un nœud de l'expérience.

9.3.5. Bilan

Nous avons présenté un ensemble de modèles dédiés à une expérience de télé-TP en informatique. Chacun de ces modèles contribue à la réalisation des buts d'une expérience, et par conséquent contribue à l'atteinte des objectifs pédagogiques de l'activité de télé-TP qui englobe cette expérience :

- Le modèle de gestion des laboratoires locaux ou distants offre une vue explicite des ressources disponibles ainsi que leur localisation, permettant à l'enseignant/concepteur de choisir celles qui correspondent le mieux à son scénario pédagogique. D'autre part, le modèle des ressources facilite le développement d'applications de gestion efficaces capables de considérer le système institutionnel de gestion des rotations et planification des séances de cours, de travaux dirigés et de travaux pratiques.

- Le modèle de gestion d'une expérience est destiné à être utilisé, au pire directement par le concepteur initié et au mieux via des applications de conception graphiques intuitives, pour faciliter l'étape initiale de modélisation des expériences. Le modèle proposé est assez complet pour construire des expériences informatiques d'une complexité arbitraire, mais sa puissance réside dans son potentiel d'extension qui permet aux concepteurs assistés (ou non) par des experts, d'intégrer des classes modélisant d'autres composants informatiques matériels (comme les périphériques) ou logiciels (comme les bases de données et autres outils).
- Enfin, le modèle des activités permet de tracer de manière fine et structurée, toute action et opération survenue sur l'expérience, qu'elle soit directement issue d'un acteur humain (sous la forme d'une commande) ou produite par le système suite à une action d'un utilisateur. Nous nous sommes limités dans un premier temps à modéliser une seule catégorie d'interaction « Homme-Expérience », celle des interfaces dites « ligne de commande », même si la représentation d'autres types d'interaction comme les interfaces graphiques ordinaires ou évoluées sont envisageables.

Nous exposons dans la suite de ce chapitre l'implémentation des différentes couches supports à la mise en œuvre de l'expérience type, en commençant par la couche d'Apprentissage. Nous nous concentrons sur les outils relatifs aux télé-TPs qui ont été intégrés dans notre cadre de travail d'un point de vue technique, le point de vue pédagogique faisant l'objet du chapitre suivant.

9.4. Implémentation de la couche d'Apprentissage

La couche d'Apprentissage est fondée sur un LMS modulaire, étendu par des outils spécifiques dédiés aux télé-TPs ; la Figure 9-13 donne un aperçu détaillé de la couche d'Apprentissage.

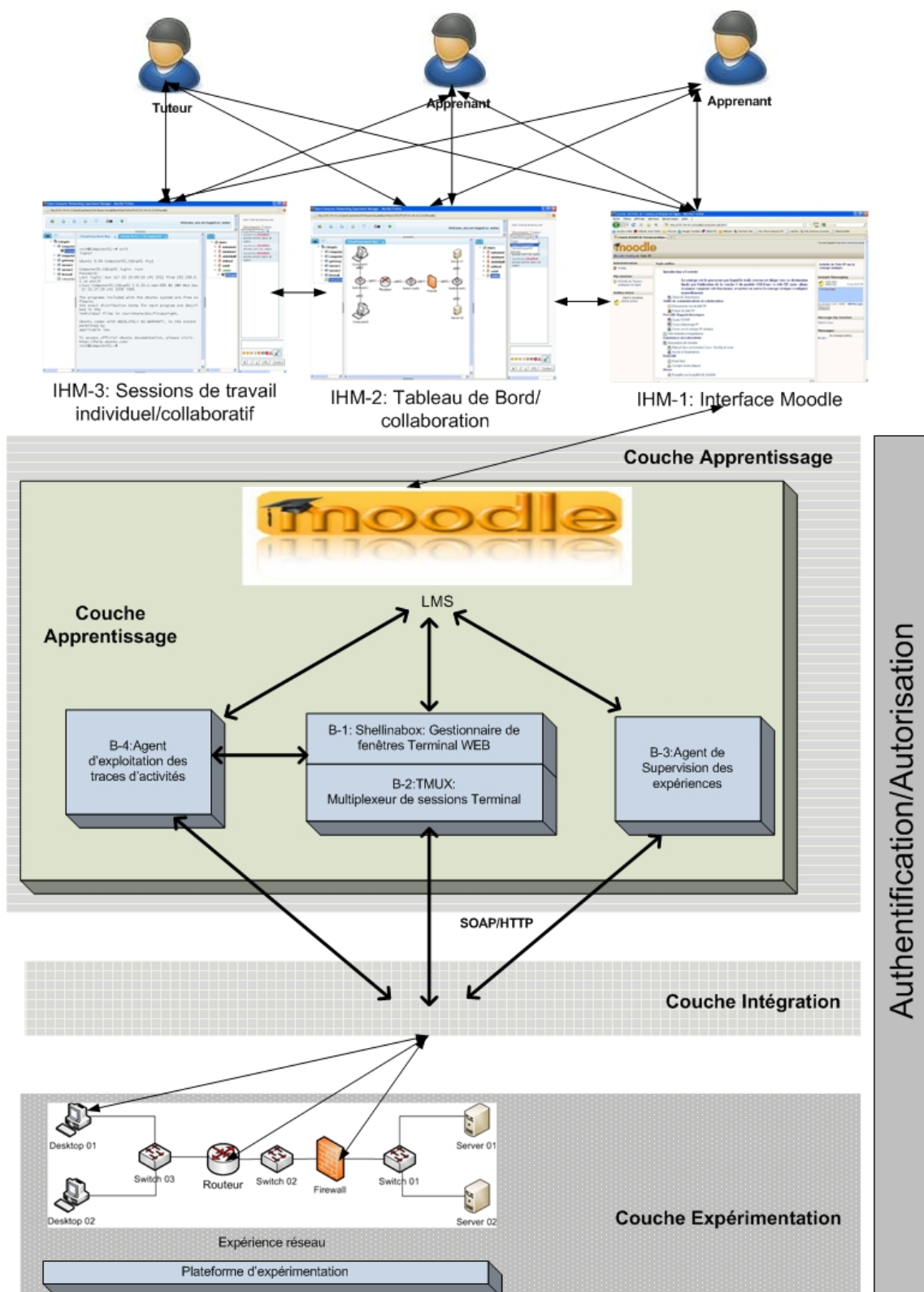


Figure 9-14. Composants de la couche d'Apprentissage.

9.4.1. La plateforme d'apprentissage

Au niveau de la couche d'Apprentissage nous avons adopté le LMS Moodle car c'est un environnement de télé-enseignement *open source* largement déployé et utilisé dans le monde, et notamment en France. Ce système de gestion d'apprentissage développé en PHP avec une base de données SQL facilite l'intégration et l'adaptation de différents types de contenus et applications d'apprentissage, en plus d'être doté de nombreux atouts tels que la gestion administrative et pédagogique de l'apprentissage ou son ouverture et sa capacité d'intégration de nouveaux modules spécialisés. De plus, la plateforme Moodle est mise en œuvre au sein de notre université, où des extensions spécifiques à la gestion des objets pédagogiques et des traces d'activités ont été développées (Broisin, 2006). Notre objectif est d'enrichir l'environnement existant en lui greffant les extensions spécifiques à la gestion d'expériences en ligne. Ceci nous permettra de montrer la validité de notre approche indépendamment du LMS cible, mais en même temps de préserver les investissements en recherche et développement effectués sur ce LMS particulier.

Cependant Moodle n'est nativement pas pourvu de fonctionnalités relatives aux télé-TPs. Alors pour permettre une gestion efficace de ce type d'activité et être en mesure de construire une IHM complexe dédiée aux expériences en ligne, nous avons implémenté le moteur d'exécution des expériences présenté dans la section 5.2.2 qui offre les fonctionnalités de base nécessaires au développement d'outils et IHMs de haut niveau facilitant l'accès et l'exploitation des expériences ; ces outils et IHMs destinés aux utilisateurs et localisés au-dessus du moteur d'exécution sont détaillés dans le chapitre 10.

9.4.2. Le moteur d'exécution des expériences

Le moteur d'exécution des expériences illustré par la Figure 9-14 est composé de deux agents (voir section 5.2.2) :

- L'agent de pilotage des expériences qui implémente les fonctions de contrôle (bloc B-1) et de supervision (bloc B-3) des expériences d'une part, et de gestion des sessions (bloc B-2) d'autre part.
- L'agent de gestion des traces d'activités sur les expériences (bloc B-4).

9.4.2.1. L'agent de pilotage des expériences

Afin d'assurer le pilotage des expériences informatiques, nous avons choisi d'implémenter, dans un premier temps, une interface de commande fondée sur les terminaux textuels dont nous avons présenté un modèle descriptif dans la section 9.3.4.

Notre objectif est d'offrir une interface de contrôle en mode ligne de commande accessible via un navigateur web afin de l'intégrer au LMS de manière transparente et de libérer les utilisateurs de toute autre installation logicielle additionnelle. Nous avons identifié différents outils *open source*, fondés sur les technologies du web 2.0, remplissant cette fonctionnalité : Shellinabox⁴⁸, Anyterm⁴⁹ et Ajaxterm⁵⁰. Bien que ces outils soient équivalents d'un point de vue utilisateur puisqu'ils offrent la sensation d'utiliser un Terminal classique, notre choix s'est porté sur l'outil Shellinabox pour la facilité et la souplesse de son intégration avec d'autres applications web. L'interface de cet outil se présente sous la forme d'un gestionnaire de Terminals textuels, que nous avons adapté selon nos objectifs (réalisme, efficacité) et contraintes liées à la distance (ressources, bande passante, etc.). Même si l'outil Shellinabox devait instancier un client SOAP dédié à la communication avec la couche d'Intégration, à ce stade du développement nous nous sommes contentés d'invoquer directement les services de la couche d'Intégration en mode client/serveur classique.

Ni Shellinabox, ni les autres outils étudiés, n'assurent cependant une gestion efficace des sessions puisqu'un dysfonctionnement du programme ou une déconnexion va détruire le travail déjà effectué. D'autre part, un utilisateur nécessite souvent la reprise d'une session existante pour réutiliser son historique ou retrouver ses variables. Aussi, dans un contexte éducatif où les durées des activités de télé-TP sont limitées, la perte d'une session Terminal ne peut être tolérée. Pour pallier ce problème, nous avons intégré un gestionnaire de sessions sophistiqué reposant sur l'outil TMUX⁵¹, principalement pour des critères de « *programmabilité* » satisfaisants par rapport à son ancêtre SCREEN⁵². TMUX offre un

⁴⁸ <http://code.google.com/p/shellinabox/>

⁴⁹ <http://anyterm.org/>

⁵⁰ <http://antony.lesuisse.org/software/ajaxterm/>

⁵¹ <http://tmux.sourceforge.net/>

⁵² <http://www.gnu.org/software/screen/>

contrôle manuel ou automatique des sessions ouvertes par les utilisateurs, et propose des capacités de disposition et d’affichage des fenêtres plus riches et plus souples, en complément d’une barre d’état. Notre gestionnaire de sessions assure alors les fonctions suivantes :

- La persistance d’une session Terminal : une session demeure ouverte même si la fenêtre Terminal dysfonctionne ou si la connexion réseau échoue. L’utilisateur peut ainsi reprendre ses sessions par le simple rafraîchissement de son interface, ou lors de sa prochaine connexion. Cette fonctionnalité limite l’inflation du nombre de sessions ouvertes, et par conséquent les traces associées sont regroupées pour une meilleure exploitation a posteriori.
- Le multifenêtrage : plusieurs fenêtres Terminal peuvent être rattachées à la même session Terminal pour y exécuter des tâches différentes.
- Le partage de sessions : plusieurs utilisateurs peuvent accéder à une même session Terminal pour travailler de façon collaborative (voir le prochain chapitre). Un utilisateur peut accéder à une session ouverte par un autre utilisateur soit en mode lecture seule, soit en mode lecture et écriture pour que chacun puisse intervenir et voir les opérations des autres.

Enfin, le pilotage des expériences doit être complété par une IHM d’exploration et de supervision des propriétés et de l’état de ces expériences, à la manière du tableau de bord d’un cockpit. Les différents indicateurs affichés par ces outils orientent en effet les actions des acteurs et augmentent leur efficacité en offrant une vue de l’expérience globale et/ou de chacun de ses composants. Dans notre implémentation, l’agent de supervision des expériences (voir bloc B-3 sur la Figure 9-14) fait appel aux services de gestion de la couche d’Intégration présentés dans la section 6.1.2 et qui offrent deux catégories de services : (1) les services d’exploration et de découverte des expériences (identification des composants, reconstitution de la structure de l’expérience, etc.) pour connaître l’évolution de l’état des composants de l’expérience, et (2) les services de gestion de sessions Terminal sur l’expérience. Ces services manipulent les classes et instances du modèle spécifique à la gestion des expériences présenté dans la section 9.3.2, et sont implémentés par un ensemble de scripts PHP accessibles aux outils et IHMs de plus haut niveau.

9.4.2.2. L'agent d'exploitation des traces d'activités

L'agent d'exploitation des traces d'activités est utilisé par l'interface utilisateur pour diverses tâches de consultation, d'analyse et de manipulation des traces d'activités enregistrées par l'outil de contrôle des sessions d'activités. Au niveau actuel du développement, cet agent communique directement avec la couche d'Intégration pour interroger la base de connaissances afin d'insérer ou de consulter des traces dans le but d'alimenter les IHMs adéquates. Les processus de cet agent sont dirigés par les modèles de gestion et de supervision des activités détaillées dans la section 9.3.4, c'est-à-dire que les objets CIM enregistrés sont des instances des classes et associations de ce modèle.

Par ailleurs, cet agent interagit avec les outils Shellinabox et TMUX afin d'offrir la possibilité de rejouer et de consulter en temps réel ou en différé des sessions de travail.

9.5. Implémentation de la couche d'Intégration

L'implémentation de la couche d'Intégration que nous avons réalisée comprend les composants décrits dans le chapitre 6, mais ne comporte que le niveau supérieur de cette couche. En effet, dans le cadre de notre expérimentation présentée dans le chapitre 11, le laboratoire d'expériences était constitué d'un banc d'essais unique ne nécessitant pas de laboratoires additionnels distants. La Figure 9-15 illustre notre architecture de la couche d'Intégration.

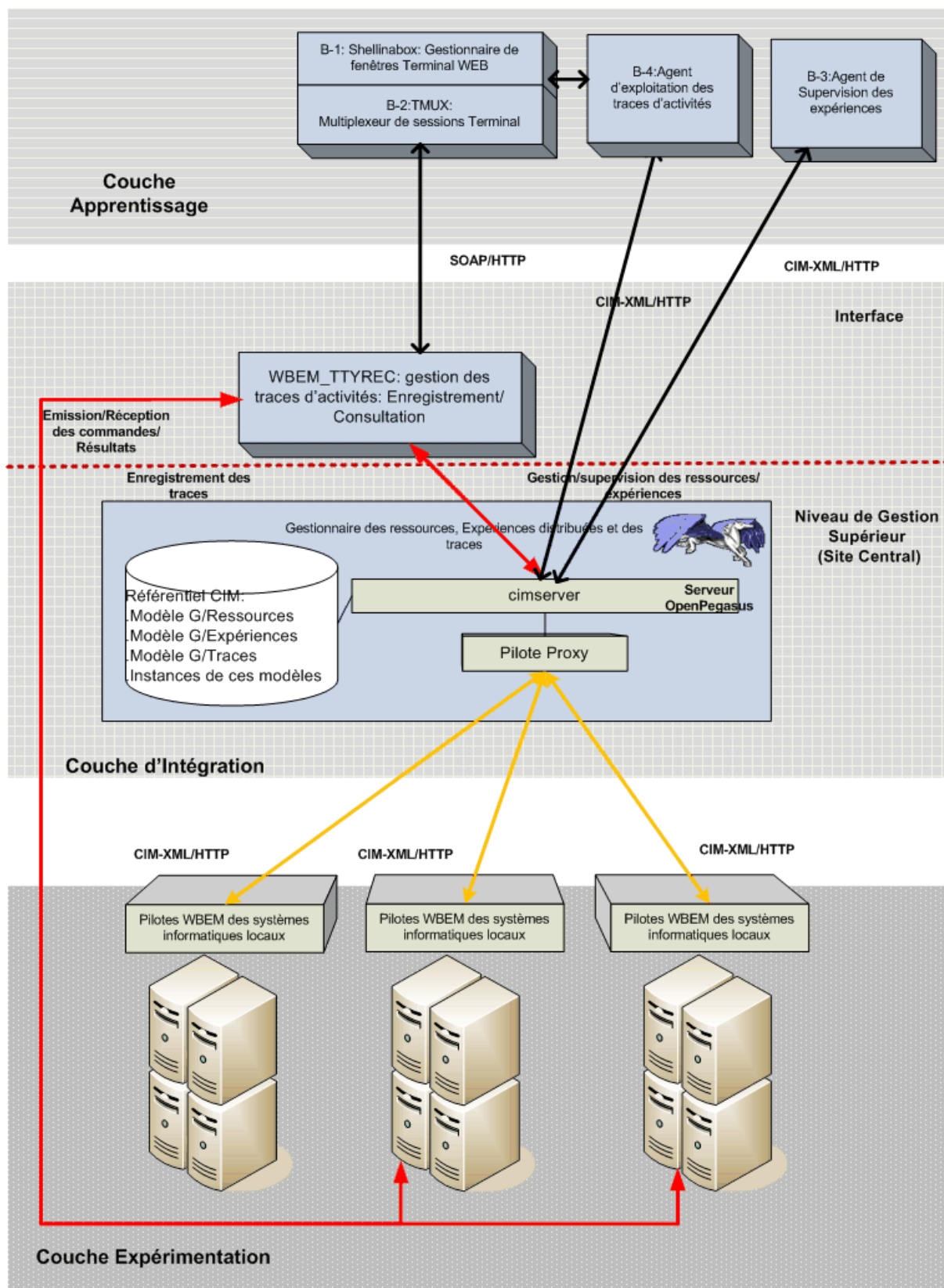


Figure 9-15. Composants de la couche d'Intégration.

9.5.1. Le serveur WBEM OpenPegasus

Pour implémenter le serveur WBEM de la couche d'Intégration, nous avons opté pour l'environnement existant OpenPegasus⁵³. Il s'agit d'un outil *open source* développé en C/C++ et maintenu par l'OpenGroup⁵⁴ qui offre une importante interface de programmation aux développeurs, jouissant ainsi du support d'une importante communauté. Ce choix est motivé non seulement par les performances de cet outil par rapport aux autres implémentations WBEM (Yoo et al., 2006), mais aussi par les fonctionnalités avancées qui sont offertes, notamment sa compatibilité avec des *Providers* WBEM tiers comme ceux du projet SBLIM⁵⁵ d'IBM, ou la possibilité d'interagir avec des *Providers* distants embarqués sur les ressources à gérer.

OpenPegasus fournit un référentiel CIM et le gestionnaire associé, alors à partir des modèles UML élaborés dans la section 9.3, nous avons produit les fichiers MOF associés avant de les injecter dans OpenPegasus. Dès lors, le logiciel est capable de manipuler les classes et instances relatives aux télé-TPs en informatique. Dans notre implémentation, OpenPegasus est responsable de l'allocation des ressources requises pour notre expérience, à son déploiement et à sa supervision, mais il n'a pas pour charge la gestion des traces d'activités des acteurs humains ; cette tâche est déléguée à un composant de substitution présenté ci-après.

9.5.2. Les services rendus à la couche d'Apprentissage

Dans la section 6.1, nous avons présenté les différents services qui doivent être proposés par la couche d'Intégration aux environnements d'apprentissage. Cependant, pour des raisons de gain de temps et faute de moyens humains, la communication entre la couche d'Apprentissage et le serveur WBEM OpenPegasus est assurée différemment dans le cadre de notre implémentation :

- L'enregistrement au sein du serveur OpenPegasus des activités réalisés par les utilisateurs, ainsi que leur opérationnalisation sur la couche d'Expérimentation, sont assurés par un

⁵³ <http://www.openpegasus.org>

⁵⁴ <http://www.opengroup.org/>

⁵⁵ http://sourceforge.net/apps/mediawiki/sblim/index.php?title=Main_Page

outil développé dans le cadre de ce travail appelé WBEM_TTYREC ; ce dernier communique avec les couches d'Apprentissage et d'Expérimentation à travers le protocole SSH.

- Les agents d'exploitation des traces d'activités et de supervision des laboratoires/ressources, qui embarquent un client WBEM fourni par l'OpenGroup, communiquent directement avec le serveur OpenPegasus via le protocole CIM/XML encapsulé dans HTTP.

Le programme WBEM_TTYREC joue donc un rôle pivot dans notre implémentation :

- Il communique avec la couche d'Apprentissage, et plus spécifiquement avec l'outil TMUX, pour intercepter les actions réalisées par les utilisateurs.
- Il communique avec la couche d'Expérimentation pour opérationnaliser ces commandes sur les nœuds de l'expérience ; il reçoit également les messages produits par les nœuds de l'expérience en réponse aux commandes émises.
- Il enregistre, au sein du serveur OpenPegasus et grâce au client WBEM qu'il embarque, ces commandes (ainsi que les retours des nœuds de l'expérience) sous la forme d'instances des classes du modèle présenté dans la section 9.3.4.

L'outil WBEM_TTYREC est fondé sur le logiciel existant TTYREC⁵⁶ dédié à l'enregistrement et à la lecture de sessions Terminal. Lorsque ce programme est invoqué, il s'exécute en arrière plan pour intercepter les caractères frappés par l'utilisateur ou générés par le système, qu'il enregistre dans un fichier texte ; un enregistrement contient les caractères entrés par l'utilisateur ou générés par le système, ainsi que la date et l'heure précises de leur production. L'ensemble des enregistrements offre alors l'opportunité de rejouer fidèlement une session d'apprentissage dans son intégralité. Nous avons donc modifié le code source de cet outil pour produire notre composant WBEM_TTYREC. Ce dernier enregistre les traces au format CIM dans le serveur OpenPegasus au lieu de les enregistrer dans un fichier au format texte. Les instances CIM, conformes à notre modèle de traces d'activités, contiennent alors toutes les informations nécessaires à la reconstitution des sessions. En effet, lorsqu'un utilisateur ouvre une session de travail sur un nœud particulier de l'expérience, un objet de type *TEL_TTYSession* modélisant la session est créé, ainsi que toutes les relations avec l'objet

⁵⁶ <http://0xcc.net/ttyrec/>

CIM_identity représentant l'utilisateur qui a initié la session. D'autre part, les relations entre l'instance *TEL_TTYSession* et l'objet *TEL_ComputerSystem* décrivant le nœud sur lequel la session est ouverte sont également mises en place, offrant ainsi l'opportunité de retrouver l'instance de l'expérience (de type *TEL_ComputerExperiment*) associée. A partir de ce moment, à chaque fois que l'utilisateur tape une commande, elle est encapsulée dans un objet CIM de type *TEL_CommandExperimentActivity* (voir la section 9.3.4) relié à l'instance *TEL_TTYSession* (présentée plus haut) à travers l'association *TEL_ActivityOnResourceComponent*. Ainsi, un EIAH peut, à partir des identifiants d'une expérience, d'un utilisateur, d'un nœud et d'une session de travail, parcourir le flux des objets CIM retraçant les commandes entrées par l'utilisateur et les réponses transmises par le système.

9.6. Implémentation de la couche d'Expérimentation

Rappelons que la couche d'Expérimentation réunit toutes les ressources mises à disposition par tous les laboratoires (d'un même établissement ou d'instituts partenaires) et pouvant être impliquées dans une expérience en ligne. Pour les besoins des télé-TPs en informatique, nous avons opté pour une plateforme d'expérimentation fondée sur les technologies de virtualisation et dotée d'un outil de gestion d'expériences. La couche d'Expérimentation résultante est illustrée par la Figure 9-16 et se décompose en trois niveaux :

- Un niveau inférieur qui a pour charge la gestion d'une expérience monoposte virtuelle, c'est-à-dire qui nécessite une machine virtuelle unique.
- Un niveau intermédiaire fondé sur un outil de virtualisation capable de gérer des expériences réseaux distribuées ou non, et constituées de plusieurs machines virtuelles.
- Un niveau supérieur constitué de Pilotes WBEM qui assure la communication avec le gestionnaire des expériences (le serveur OpenPegasus) localisé dans la couche immédiatement supérieure (la couche d'Intégration).

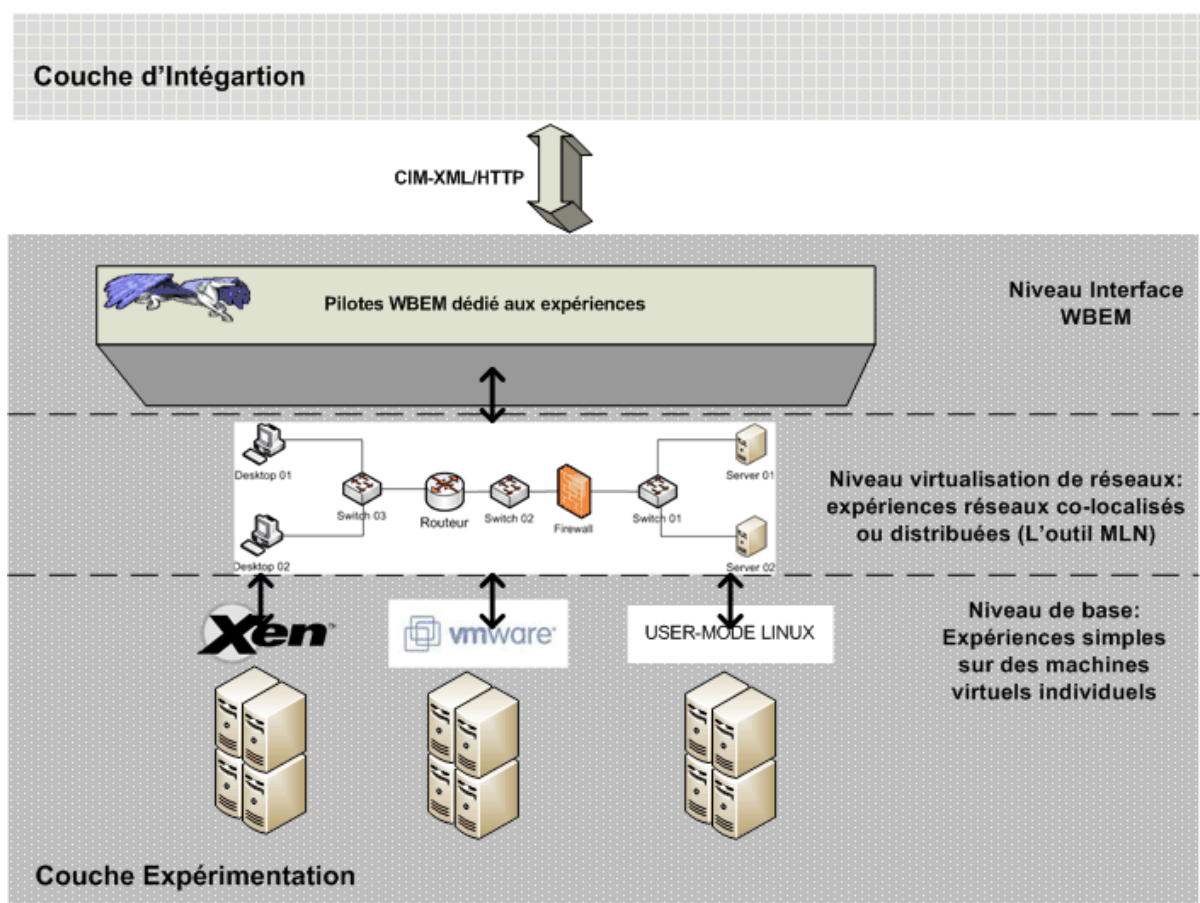


Figure 9-16. Niveaux et composants de la couche d'Expérimentation.

9.6.1. Le niveau inférieur

La technologie de virtualisation dans notre contexte fait référence à la virtualisation des systèmes informatiques qui permet l'exécution, sur une machine physique appelée *machine hôte*, d'une ou de plusieurs machines virtuelles appelées *machines invitées*. Chaque machine invitée possède ses propres ressources matérielles⁵⁷, son système d'exploitation et ses applications, indépendamment de la machine hôte et des autres machines virtuelles.

Le choix de cette technologie se justifie par le fait qu'elle apporte une alternative intéressante aux laboratoires physiques et aux simulations, en combinant leurs avantages et en minimisant leurs inconvénients. En effet, la technologie de virtualisation favorise l'atteinte de l'efficacité pédagogique grâce à sa fidélité technique et au sentiment de réalisme qu'elle procure à

⁵⁷ Ressources matérielles virtualisées

l'apprenant en assurant la conception d'expériences aussi réalistes et complexes que les expériences traditionnelles. Aussi, elle est dotée de capacités de contrôle des paramètres des expériences et de gestion du cycle de vie des machines virtuelles (conception, construction, démarrage, exploitation, arrêt, sauvegarde, restauration et suppression). Enfin, notre choix est renforcé par le fait que ces nouvelles technologies ne sont plus seulement utilisées pour les tests, mais également introduites dans le monde industriel afin de minimiser les coûts d'acquisition et d'exploitation des ressources informatiques, et de favoriser leur mutualisation.

Différentes technologies de virtualisation existent, toutes fondées sur le même principe : une couche intermédiaire assure la communication entre les machines virtuelles et la machine physique qui les héberge. Les différences entre les diverses technologies de virtualisation existantes résident principalement au niveau de cette couche intermédiaire et des propriétés des machines virtuelles qui s'y exécutent. Nous ne donnons pas ici les détails de ces technologies - le lecteur est invité à consulter certains guides techniques (Donnette et Hannequin, 2007 ; Smile, 2010) - mais le Tableau 9-2 en présente une comparaison succincte.

Tableau 9-2. Comparaison des différentes technologies de virtualisation.

Type de technologie de virtualisation	Propriétés de la couche de virtualisation	Exemples d'outils
Emulation matérielle	<ul style="list-style-type: none"> - Emulation complète d'une plateforme matérielle - Faibles performances dues à la traduction des instructions de bas niveau entre le système hôte et le système invité 	BOCHS ⁵⁸ QEMU ⁵⁹
Virtualisation complète	<ul style="list-style-type: none"> - Fondée sur une machine virtuelle médiatrice (ou hyperviseur) chargée de gérer la communication entre les systèmes invités et le système hôte - Permet de déployer des systèmes hôtes non modifiés - Plus rapide que l'émulation matérielle, mais 	VMWARE ⁶⁰ IBM z/VM ⁶¹ KVM ⁶² XEN ⁶³

⁵⁸ <http://bochs.sourceforge.net/>

⁵⁹ http://wiki.qemu.org/Main_Page

⁶⁰ <http://www.vmware.com>

⁶¹ <http://www.vm.ibm.com/>

	performances moyennes	
Paravirtualisation	<ul style="list-style-type: none"> - Fondée sur le principe de l'hyperviseur - Nécessite la modification du système invité pour coopérer avec le système hôte - Performances excellentes 	CITRIX XenServer XEN (Barhms et al., 2003) User-Mode Linux ⁶⁴
Isolation de systèmes d'exploitation	<ul style="list-style-type: none"> - La couche de virtualisation se situe au dessus du système d'exploitation du système hôte - Le système invité est isolé dans un contexte d'exécution, tel un processus - Nécessite la modification du système invité pour coopérer avec le système hôte - Performances excellentes 	Parallels ⁶⁵ Linux V-Server ⁶⁶ OpenVZ ⁶⁷ Virtuozzo ⁶⁸

A partir des différentes études comparatives entre les différentes technologies de virtualisation existantes et les divers outils associés (voir plus loin), nous avons choisi la technologie XEN qui assure un bon compromis entre performances techniques, coût, maturité et stabilité (Smile, 2010 ; Donnette et Hannequin, 2007).

Notons que le cycle de vie d'une expérience monoposte est associé à celui de la machine virtuelle dont les différentes étapes figurent dans le Tableau 9-3.

Tableau 9-3. Phases du cycle de vie d'une machine virtuelle.

Phase	Description
Modélisation	Consiste à renseigner les caractéristiques matérielles et logicielles de la machine virtuelle pour pouvoir la construire. Chaque technologie possède son propre langage,

⁶² http://www.linux-kvm.org/page/Main_Page

⁶³ <http://www.xen.org>

⁶⁴ <http://user-mode-linux.sourceforge.net/>

⁶⁵ <http://www.parallels.com/>

⁶⁶ http://linux-vserver.org/Welcome_to_Linux-VServer.org

⁶⁷ http://wiki.openvz.org/Main_Page

⁶⁸ <http://wiki.openvz.org/Virtuozzo>

	mais récemment un standard a été proposé par le DMTF appelé <i>Open Virtual Format</i> (OVF ⁶⁹) pour unifier ces langages de modélisation.
Construction	Construction du système de fichiers de la machine virtuelle sur la base de son modèle, après la vérification syntaxique et sémantique de ce dernier.
(Re)démarrage, Restauration	Allocation des ressources nécessaires à la machine (processeur, mémoire et périphériques virtuels) avant son exécution effective.
Exploitation	Utilisation des outils de télé-opération (Console, SSH, VNC, etc.) pour accéder en contrôle sur la machine virtuelle.
Arrêt (Suspension)	Arrêt de l'exécution de la machine virtuelle et libération des ressources allouées.
Archivage	Sauvegarde du système du modèle et du fichier de la machine virtuelle dans un état stable donné pour une éventuelle restauration dans le même état.
Suppression	Suppression du modèle et du système de fichier de la machine virtuelle ainsi que de toutes ses traces.

Enfin, chaque technologie de virtualisation utilise une technologie de commutateurs virtuels qui permet d'interconnecter la machine virtuelle au monde extérieur ou plusieurs machines virtuelles entre elles. De telles technologies rendent possibles la construction de réseaux complexes de machines virtuelles, simulant les réseaux réels. Comme exemple de technologies, citons la technologie VNET de VMWARE, la technologie *open source* Linux BRIDGE utilisée par XEN, ou la technologie UML_SWITCH utilisée par User-Mode Linux.

9.6.2. Le niveau intermédiaire

A la différence du niveau inférieur, le niveau intermédiaire doit supporter la gestion d'expériences virtuelles nécessitant un ensemble de machines physiques et/ou virtuelles. Le cycle de vie d'une expérience réseau est alors associé à celui des machines virtuelles qui la composent (une expérience réseau ne peut démarrer avant le démarrage des nœuds qui la constituent, et elle ne peut être arrêtée qu'après l'arrêt de ces mêmes nœuds), et des mécanismes complexes doivent être mis en œuvre pour éviter des situations incohérentes.

⁶⁹ <http://www.vmware.com/appliances/getting-started/learn/ovf.html>,
<http://www.dmtf.org/standards/mgmt/vman/>

9.6.2.1. Les critères de sélection

Il existe un grand nombre de technologies et d'outils matures caractérisés par des propriétés différentes pour atteindre la construction d'expériences réseau complexes reposant sur une technologie de virtualisation sous-jacente. Même si ces systèmes épargnent un long travail de développement et assurent une stabilité certaine des outils, il nous a fallu faire un choix en fonction des critères de sélection suivants :

- *Ouverture du langage de modélisation* (en format texte ou XML par exemple) : plus le format est simple et ouvert, plus il facilite le développement d'interfaces d'édition graphiques et de traduction entre le format de l'outil et le langage CIM/WBEM.
- *Exhaustivité* : le degré de couverture du plus grand nombre de situations possibles, des plus simples aux plus complexes, est un facteur important dans notre choix.
- *Diversité des technologies* : le support du plus grand nombre de technologies de virtualisation afin de profiter des avantages de chacune et de pouvoir diversifier les scénarios d'expériences.
- *Simplicité* : au niveau de son installation et de son usage.
- *Performances* : il s'agit d'éviter les frustrations et les mauvaises interprétations dues à la lourdeur de l'accès ou à l'exécution d'opérations sur l'expérience distante.
- *Gestion complète du cycle de vie d'une expérience*.
- *Gestion d'expériences distribuées* : avoir la possibilité de déployer des expériences distribuées sur plusieurs machines physiques soit parce que l'expérience est très complexe, soit parce que les ressources d'un site ne sont pas suffisantes. C'est un critère important pour la mise en œuvre des principes de partage et de mutualisation de ressources.
- *Interface graphique web* : afin de faciliter la conception d'expériences et d'unifier l'interface utilisateur accessible via le LMS (Moodle dans notre cas), il est préférable de disposer d'une interface graphique orientée web pour la modélisation des expériences.
- *Extensibilité* : avoir la possibilité d'inclure des fonctionnalités non disponibles par défaut.

9.6.2.2. Etude des outils de gestion d'expériences réseau virtuelles

En fonction des critères établis ci-dessus, nous avons étudié un panel d'outils de gestion d'expériences réseau virtuelles (Rimondini et al., 2007) : VNUML⁷⁰, NETKIT⁷¹, GINI⁷², VNE⁷³, Marionnet⁷⁴, XenWorlds⁷⁵, MLN⁷⁶, VMWARESERVER et VNEC⁷⁷. Le Tableau 9-4 expose la projection des critères sur cette liste d'outils.

Tableau 9-4. Projection des critères de sélection sur des outils de gestion d'expériences réseau virtuelles.

Outil	Technologie virtuelle sous-jacente	Format du modèle	IHM	Expériences distribuées	Options avancées
VNUML	UML uml_switch	XML	Via VNUML- GUI	NON	L'outil EDIV permet de construire des expériences VNUML distribuées.
NetKit	UML uml_switch	Texte et XML	Via NetGUI et NetEdit	NON	L'outil AutoNetKit permet de construire des expériences NetKit distribuées.
XenWorlds	XEN LinuxBridge	Texte	NON	NON	
VMWARE Server	VMWARE / VMWARE VNET	Visuel	OUI	NON	Les machines virtuelles peuvent être reconfigurées en format VMDK (textuel) ou XML/OVF.
VNEC	VMWARE / VMWARE		OUI	NON	L'interface TimeLine permet de préconfigurer des actions à

⁷⁰ http://www.dit.upm.es/vnumlwiki/index.php/Main_Page

⁷¹ http://wiki.netkit.org/index.php/Main_Page

⁷² <http://www.cs.mcgill.ca/~anrl/projects/gini/>

⁷³ <http://ndl.uva.netherlight.nl/trac/vne>

⁷⁴ <http://www.marionnet.org/>

⁷⁵ <http://home.eng.iastate.edu/~hawklan/xw-index.html>

⁷⁶ <http://mln.sourceforge.net/>

⁷⁷ <http://vnec.sourceforge.net/>

	VNET				exécuter automatiquement.
Marionet	UML VDE	Texte	OUI	OUI	L'outil VDE est utilisé pour interconnecter des machines sur des réseaux Ethernet différents.
GINI	UML	XML	Via gBuilder	OUI	Emulation de routeurs Cisco et d'équipements Wireless.
VNE	UML	Texte	NON	NON	
MLN	UML uml_switch XEN/bridge VMWARE VMWARE VNET	Texte	NON	OUI	MLN est extensible via son système de <i>plugins</i> . Des <i>plugins</i> sont disponibles sur le site de l'outil pour le support de fonctionnalités avancées. D'autres technologies de virtualisation peuvent être supportées, telles que KVM à partir du <i>plugin</i> adéquat. Le développement de <i>plugins</i> est facilité.

Nous constatons que la plupart des outils analysés proposent un langage de modélisation simple et ouvert permettant de définir les nœuds de l'expérience ainsi que les liens entre eux, et même parfois d'autres caractéristiques comme les paramètres de configuration de chaque nœud (système d'exploitation, paramètres réseau, etc.) ou les paramètres de contrôle des liens (bande passante, etc.). Beaucoup d'outils (VMWARE, VNEC, Marionet, et GINI) fournissent une interface graphique associée au langage de modélisation, d'autres technologies comme VNUML et NETKIT peuvent être complétées par des outils tiers, alors que MLN et VNE ne sont pas munis de telles capacités. Malheureusement, les interfaces disponibles sont des applications clientes autonomes et locales, souvent spécifiques à un système d'exploitation et qui sont difficilement réutilisables. Un développement est donc nécessaire à ce niveau pour offrir aux utilisateurs l'opportunité de facilement concevoir une expérience réseau virtuelle, comme nous le verrons dans le prochain chapitre.

Au niveau des performances, les outils dépendent de la technologie de virtualisation sous-jacente. Des études comparatives classent la technologie XEN en première place, suivie de User-Mode Linux puis VMWARE Server (Walter et al., 2007). Les outils XENWORLDS et MLN figurent par conséquent parmi les plus performants.

Du point de vue de l'exhaustivité, la technologie VMWARE est bien placée car elle permet d'installer des systèmes non modifiés et de créer des expériences réseau mélangeant différents systèmes d'exploitation (Linux, Windows, etc.). La technologie XEN ne permet cela que dans une architecture matérielle spécifique à base de processeur INTEL 64 bits VT ou AMD 64 bits AM-V. Par rapport à ce critère, l'outil MLN est le mieux placé car il supporte nativement trois technologies de virtualisation, en plus du support de technologies additionnelles via des *plugins* spécifiques.

Les outils MLN, GINI et Marionet supportent nativement les expériences distribuées, mais des expériences distribuées avec VNUML et NetKit sont envisageables via des outils tiers.

Enfin, seul l'outil MLN permet sa propre extension grâce à son système de *plugins*. Un *plugin* correspond à une fonctionnalité d'automatisation de tâches (installation, configuration d'un logiciel ou d'un service, etc.) pour une expérience, ou au rajout d'une fonctionnalité avancée à l'outil.

9.6.2.3. Notre choix technologique

Cette étude nous a conduit à choisir l'outil MLN pour l'émulation d'expériences réseau virtuelles car il assure le meilleur compromis entre les différents critères de sélection. En particulier, les critères de diversité, de performances, de distribution et d'extensibilité sont les points forts de cet outil. Notre choix est renforcé par le fait que cet outil a été adopté dans le projet SOFTICE (softice, 2008) dédié à l'enseignement de l'informatique. MLN offre une souplesse pour supporter diverses technologies de virtualisation sous-jacentes, et facilite le passage d'une expérience d'une technologie vers une autre grâce à son langage de modélisation commun. Par ailleurs, MLN assure une haute disponibilité des expériences distribuées puisque la technologie XEN permet de faire migrer une machine virtuelle d'une machine physique vers une autre pour obtenir de meilleures performances ; une présentation

détaillée de cet outil et de son langage de modélisation peut être trouvée sur le site Internet du projet (MLN, 2009).

La configuration précise de notre expérience type selon le langage MLN est présentée dans l'annexe E, alors que l'algorithme de traduction du format CIM-MOF vers le format MLN est détaillé dans le chapitre 10.

9.6.3. Le niveau supérieur : les Pilotes WBEM

Le niveau supérieur de la couche d'Expérimentation assure deux fonctions principales : il reçoit les messages transmis par la couche d'Intégration au format WBEM, et communique avec l'outil MLN de gestion des expériences réseau pour opérationnaliser ces instructions. Ce niveau doit donc être pourvu de Pilotes WBEM implémentant les modèles présentés dans la section 9.3, alors que l'outil MLN doit être adapté pour satisfaire les besoins de notre approche.

9.6.3.1. Implémentation des Pilotes WBEM

Fondé sur le langage C/C++, le cadre de travail SimpleWBEM⁷⁸ a été choisi pour l'implémentation des différents Pilotes WBEM de la couche d'Expérimentation car il permet un développement rapide à partir de modèles CIM exprimés au format MOF en générant automatiquement tous les fichiers entêtes et sources des différentes classes CIM. Le travail du développeur consiste alors, au niveau des méthodes générées, à insérer du code faisant appel aux commandes et bibliothèques spécifiques à la plateforme gérée. Le nouveau module logiciel qui pilote le système géré (MLN dans notre expérimentation) est ensuite déclaré auprès du serveur WBEM afin que ces entités soient en mesure d'interagir.

Dans notre implémentation, une expérience en informatique modélisée par un objet *LAB_ComputerExperiment* est associée à un projet MLN. Ainsi, les propriétés et méthodes de cette classe sont implémentées par un Pilote WBEM qui fait appel aux commandes offertes par l'outil MLN. Le Tableau 9-5 expose les commandes et paramètres de configuration d'un projet MLN utilisés pour fixer les attributs d'un objet *LAB_ComputerExperiment* et implémenter ses méthodes.

⁷⁸ <http://www.simplewbem.org>

Tableau 9-5. Correspondance entre les attributs/méthodes d'un objet *LAB_ComputerExperiment* et le projet MLN associé.

Attributs de <i>LAB_ComputerExperiment</i>	Propriété ou commande du projet MLN	Remarques
<i>ExperimentationToolIdentifier</i>	MLN	
<i>ExperimentID</i> : nom expérience	<i>project nom-expérience</i>	
<i>Deployed</i> : True/False	<i>mln status -p nom-expérience</i>	L'attribut est fixé à <i>True</i> si le projet a été construit
<i>Started</i> : True/False	<i>mln status -p nom-expérience</i>	L'attribut est fixé à <i>True</i> si le résultat est égal à <i>up</i> , et à <i>False</i> sinon
<i>Distributed</i> : True/False	<i>mln parse -f fichier-mln</i>	Fixé à <i>True</i> si l'option <i>service_host</i> est utilisée dans la configuration d'un nœud du projet (voir plus bas)
Méthodes de <i>LAB_ComputerExperiment</i>		
<i>Create()</i>	<i>mln build -p fichier-conf-mln</i>	Utilise un fichier de configuration contenant la description du projet MLN
<i>Start()</i>	<i>mln start -p nom-expérience</i>	Prend comme argument le nom de l'expérience
<i>Stop()</i>	<i>mln stop -p nom-expérience</i>	
<i>Resume()</i>	<i>mln start -p nom-expérience</i>	
<i>Save()</i>	<i>mln export -p nom-expérience</i>	
<i>Restore()</i>	<i>mln import -p nom-expérience</i>	

L'invocation d'une commande MLN par une méthode du Pilote est suivie d'un traitement consistant à affecter les valeurs des attributs de l'objet *LAB_ComputerExperiment* en fonction des résultats générés par cette commande.

En plus du Pilote dédié à la gestion d’une expérience dans sa globalité, nous avons développé deux Pilotes WBEM responsables respectivement des associations *SystemComponent* et *Component* (présentées dans la section 9.3.2.1). La première identifie les instances de type *LAB_ComputerSystem* qui constituent l’expérience, alors que la seconde retrouve l’instance unique de type *LAB_TopologyGraph* associée à l’expérience. Nous avons également développé le Pilote de l’association *MemberOfCollection* afin d’obtenir, à partir de l’instance de type *LAB_TopologyGraph*, la liste des commutateurs virtuels interconnectant les nœuds de l’expérience (de type *LANConnectivitySegment*). L’implémentation de ces relations d’association repose sur la commande *mln status -p nom-expérience* qui retrouve la liste des nœuds et des commutateurs associés à un projet MLN, ainsi que leurs états opérationnels. Par ailleurs nous avons utilisé la commande *mln parse -p nom-expérience* qui retourne toutes les propriétés d’un projet MLN à partir desquelles nous avons, entre autres, identifier le caractère distribué ou non d’une expérience.

Avant d’implémenter les autres Pilotes WBEM relatifs aux autres classes du modèle de gestion d’une expérience, nous avons recherché les Pilotes WBEM existants couvrant les classes CIM de notre modèle. Nous avons identifié dans le projet SBLIM⁷⁹ un très grand nombre de Pilotes couvrant la majorité des classes dédiées à la gestion des machines sous Linux. En particulier, nous avons réutilisé les Pilotes suivants:

- *CMPI-Base* qui implémente les classes *ComputerSystem* et *OperatingSystem*, ainsi que l’association *RunningOS*.
- *CMPI-FSVOL* est responsable de la classe *FileSystem* et de ses différentes déclinaisons (*ext2FS*, *ext3FS*, etc.), mais aussi des associations *HostedFileSystem* et *BootOSFromFS*.
- *CMPI-Network* assure la gestion des classes *NetworkPort* (ainsi que ses dérivés *EthernetPort* et *TokenRingPort*) et *IPProtocolEndPoint*, et des associations *NetworkPortImplementsIPEndpoint* et *SystemDevice*.

Nous avons ensuite développé les Pilotes des classes restant à opérationnaliser afin de compléter l’implémentation de notre modèle :

⁷⁹ http://sourceforge.net/apps/mediawiki/sblim/index.php?title=Main_Page

- L'association *LAB_SystemResource* pour retrouver, à partir de l'instance de la ressource *LAB_ComputerSystem*, l'objet *ComputerSystem* qui représente concrètement le nœud d'une expérience.
- La classe *LANEndPoint* pour gérer un port réseau logique.
- L'association *DeviceSAPImplementation* pour retrouver, à partir d'une instance de l'objet précédent, l'objet *NetworkPort* qui représente le port réseau physique associé.
- L'association *BindsToLANEndPoint* afin de lier une instance de l'objet *LANEndPoint* à l'objet *IPProtocolEndPoint* qui renseigne sur les paramètres IP (adresse, masque de sous-réseau, etc.) du port logique associé.

Ainsi, l'ensemble des Pilotes développés ou réutilisés dans le cadre de nos travaux permet de retrouver les instances des entités qui composent une expérience. L'état global d'une expérience est obtenu par la lecture des attributs des instances de tous les objets qui la constituent, et les opérations sur l'expérience sont exécutées à travers les différentes méthodes implémentées par les Pilotes de ces objets.

9.6.3.2. Adaptation de l'outil MLN

Afin de préparer une expérience à être gérée par les Pilotes WBEM présentés ci-dessus, nous avons développé des extensions et rajouté des options de configuration au modèle MLN natif d'une expérience. Ce réaménagement consiste en une série d'opérations :

- Installer, au sein du serveur physique exécutant l'outil MLN, les Pilotes WBEM des classes *LAB_ComputerExperiment*, *LAB_TopologyGraph*, *LAB_ComputerSystem* et *LANConnectivitySegment*, ainsi que les associations *SystemComponent* et *Component*.
- Doter chaque machine virtuelle d'une interface réseau dédiée à sa gestion et inaccessible aux apprenants.
- Installer, dans chaque machine virtuelle constituant l'expérience, tous les Pilotes WBEM mentionnés dans la section précédente afin de permettre le contrôle et la supervision des machines virtuelles par la couche d'Intégration.
- Créer un compte utilisateur pour l'apprenant sur chaque machine virtuelle.
- Fixer les paramètres généraux de chacune des machines virtuelles.

Nous avons énoncé à plusieurs reprises que la gestion des expériences dont les machines virtuelles sont dispersées sur différents serveurs physiques, doit être assurée par le serveur WBEM primaire de la couche d'Intégration. Dans notre implémentation, la gestion des expériences distribuées est simplifiée par le fait que l'outil MLN est doté de la capacité de gérer nativement ce genre d'expériences d'une manière centralisée, à partir d'une hiérarchisation des différentes machines exécutant MLN. Par conséquent, un seul Pilote WBEM installé au niveau du serveur MLN principal suffit pour contrôler et superviser une expérience distribuée sur plusieurs machines physiques.

9.7. Scénario de coopération des différentes couches

La Figure 9-17 illustre la position de l'ensemble des composants de notre implémentation, et décrit l'ordonnancement des étapes nécessaires à la prise de contrôle d'une machine de l'expérience :

- Lorsqu'un utilisateur connecté à Moodle souhaite ouvrir une session sur une machine virtuelle (1 et 2), le programme Shellinabox est lancé afin de fournir l'accès à cette machine (3).
- L'exécution de Shellinabox déclenche le démarrage du gestionnaire de sessions TMUX (4) qui produit un identifiant unique de session.
- TMUX se connecte au programme WBEM_TTYREC situé dans la couche d'Intégration (6) afin d'émettre la demande de connexion à la machine virtuelle.
- WBEM_TTYREC invoque alors la commande d'ouverture de session sur la machine de l'expérience ciblée via une connexion SSH qui passe par le serveur des expériences (9,11).
- WBEM_TTYREC commence l'enregistrement des traces d'activités sur le serveur WBEM (8), en se substituant au client gSOAP dans le stade actuel de développement.
- La commande SSH sur le serveur d'expériences invoque à son tour l'ouverture d'une session Terminal en mode console ou SSH sur la machine cible (12). Si cette étape réussit, alors l'utilisateur pourra commencer à travailler sur la machine.

En parallèle, l'IHM de l'expérience détaillée dans le chapitre suivant invoque certaines commandes WBEM auprès du serveur WBEM (07) afin de présenter à l'utilisateur une image

cohérente de l'état global de l'expérience et offrir l'opportunité de la superviser et de la contrôler (13).

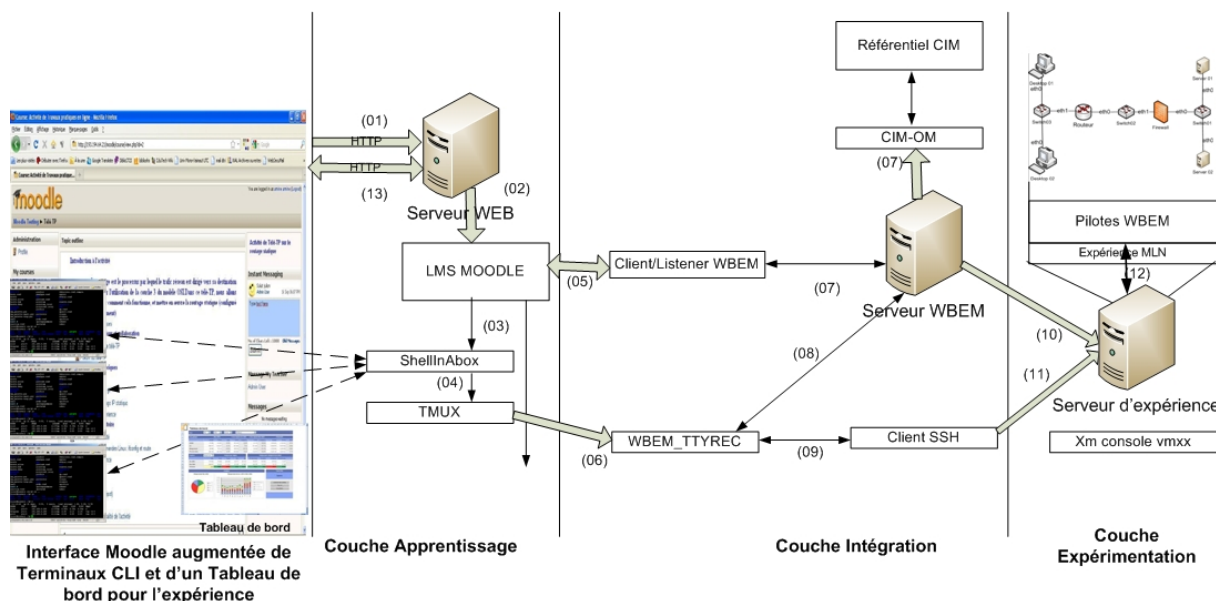


Figure 9-17. Coopération des différents outils pour contrôler une machine de l'expérience.

9.8. Synthèse

Dans ce chapitre, nous avons appliqué l'architecture générique présentée dans la partie II de ce rapport au contexte de l'enseignement de l'informatique, et implémenté ses divers composants. Une partie de cette implémentation est commune à toutes les disciplines, il s'agit de la couche d'Intégration qui renferme les processus de traitement et d'intermédiation entre les couches d'Apprentissage et d'Expérimentation.

Bien que les composants de la couche d'Intégration soient communs, les modèles CIM qui y sont implantés demeurent spécifiques à la discipline enseignée, au type de laboratoires et de ressources, mais également au type d'expériences. Le gestionnaire existant OpenPegasus est au cœur de la couche d'Intégration, alors que nous avons développé nos propres modèles CIM dédiés à la gestion des ressources, expériences et activités liées à la discipline de l'informatique.

Au niveau de la couche d'Apprentissage, nous avons opté pour le LMS Moodle que nous avons étendu avec des composants spécifiques aux activités de télé-TP. Une activité de télé-TP devient alors une activité d'apprentissage ordinaire intégrant une ressource particulière :

« l'expérience ». L'interface d'expérimentation est actuellement limitée aux sessions en mode Terminal texte, mais elle offre des IHMs interactives accessibles via n'importe quel navigateur web et aussi réalistes que celles proposées par des applications traditionnelles. Dans notre cas, elle intègre des outils *open source* puissants dont la combinaison assure une gestion efficace des expériences en informatique en mode Terminal texte en plus de l'enregistrement en arrière plan des activités réalisées par les différents acteurs humains.

Enfin, au niveau de la couche d'Expérimentation, nous avons opté pour la technologie de virtualisation XEN et le gestionnaire d'expériences réseau MLN qui assurent ensemble un bon compromis entre facilité d'usage, performances, coût, diversité des scénarios d'apprentissage et prise en compte des expériences distribuées. Toutefois, afin d'assurer la communication entre le couple MLN/XEN et la couche d'Intégration, nous avons développé les Pilotes WBEM qui opérationnalisent les modèles déclarés dans les serveurs WBEM, et apporté les extensions nécessaires à l'outil MLN pour supporter ces composants.

L'environnement de télé-TP développé dans nos travaux supporte et facilite les activités de travaux pratiques en informatique qui ne nécessitent que l'utilisation d'interfaces de type « Ligne de Commande », et peut donc être réutilisé dans le cadre de l'enseignement et de l'apprentissage d'un large éventail de disciplines telles que les systèmes d'exploitation, les réseaux et télécommunication, la sécurité, les grilles de calcul, la virtualisation, l'informatique des nuages, le développement et la programmation nécessitant des environnements de développement et de débogage en mode texte, etc.

Pour les autres disciplines fondées sur des environnements graphiques, notre environnement doit être étendu :

- Au niveau de la couche d'Apprentissage, des IHMs (de conception et d'exploitation des expériences) correspondant aux différents logiciels impliqués dans l'activité de télé-TP sont à élaborer.
- Au niveau de la couche d'Intégration, les modèles spécifiques aux activités réalisables à travers ces expériences, ainsi que ceux dédiés à leur supervision, doivent être définis.
- Au niveau de la couche d'Expérimentation, il est nécessaire d'intégrer et/ou de développer les composants informatiques spécifiques à la discipline à enseigner, ainsi que les Pilotes WBEM implémentant ces modèles spécifiques.

Chapitre 10. Outils et IHMs supports aux activités pédagogiques

<u>10.1. Introduction</u>	238
<u>10.2. Outils et IHMs supports aux activités de l'enseignant concepteur</u>	239
<u>10.2.1. Conception des expériences avec un éditeur graphique</u>	240
<u>10.2.2. Traduction vers les formats MOF et MLN</u>	243
<u>10.2.3. Gestion du cycle de vie des expériences</u>	245
<u>10.2.4. Stockage et partage des objets pédagogiques « Expérience »</u>	249
<u>10.3. Outils et IHMs supports aux activités d'apprentissage individuel</u>	249
<u>10.3.1. La barre de menus</u>	251
<u>10.3.2. L'explorateur des expériences</u>	252
<u>10.3.3. Exploitation de l'expérience</u>	256
<u>10.3.4. L'explorateur des utilisateurs</u>	259
<u>10.4. Description de l'espace d'apprentissage collaboratif</u>	261
<u>10.4.1. Le sentiment de présence</u>	261
<u>10.4.2. Exploitation de sessions ouvertes par d'autres acteurs</u>	261
<u>10.4.3. Création et exploitation de sessions partagées</u>	262
<u>10.4.4. L'outil de messagerie instantanée</u>	263
<u>10.5. Description de l'espace de travail du tuteur</u>	265
<u>10.6. Synthèse</u>	268

10.1. Introduction

Dans ce chapitre, nous présentons les interfaces offertes aux utilisateurs impliqués dans une activité de télé-TP pour le support des activités pédagogiques. Ces IHMs, développées dans le cadre de nos travaux, sont fondées sur la technologie AJAX pour exploiter les fonctionnalités du moteur d'exécution : les IHMs font appel, en arrière plan, aux opérations de base offertes

par le moteur d'exécution des expériences. Rappelons toutefois que l'architecture générale de notre EIAH est assez souple pour permettre à chacun de développer ses propres outils et interfaces, grâce à l'externalisation des informations de gestion relatives aux expériences vers le serveur WBEM.

Ce chapitre reprend l'exemple de l'activité de télé-TP type présentée dans le chapitre précédent, ainsi que l'expérience type dans le domaine des réseaux afin de faire le lien entre les différents outils de notre EIAH et les interfaces supportant certaines activités pédagogiques. Rappelons que selon notre vision, l'efficacité pédagogique des activités de télé-TP ne peut être atteinte que si les conditions suivantes sont réunies : (1) une conception adaptée des supports et du scénario d'apprentissage, (2) un support efficace à l'apprentissage individuel en ligne, (3) un support efficace au travail d'équipe en ligne, et (4) un support efficace au tutorat en ligne.

10.2. Outils et IHMs supports aux activités de l'enseignant concepteur

Dans un cursus en ligne Moodle, un enseignant concepteur a pour charge d'introduire un certain nombre de ressources et d'activités, d'affecter les apprenants à ces activités, d'associer des enseignants ayant des privilèges de modification du contenu (Enseignants) ou non (Tuteurs), et de fixer (ou non) un planning des activités. Par analogie aux autres ressources d'apprentissage, l'enseignant concepteur conçoit une activité de télé-TP par l'intégration d'une ou de plusieurs ressources d'apprentissage de type « Expérience » avant d'affecter les apprenants, les enseignants et les tuteurs à ce télé-TP.

La mise en œuvre d'une expérience est une activité de prototypage où l'enseignant concepteur suit le cycle suivant illustré par la Figure 10-1 :

- Conception graphique de l'expérience.
- Traduction du modèle graphique en un modèle d'instances CIM conforme au modèle dédié à la description des OPIs « Expérience » et exprimé au format MOF (voir section 9.3.3).
- Traduction du modèle CIM-MOF vers le modèle des expériences MLN.
- Déploiement de chaque machine virtuelle de l'expérience.

- Déploiement de l'expérience toute entière par la construction des connexions réseau entre ses différents nœuds.

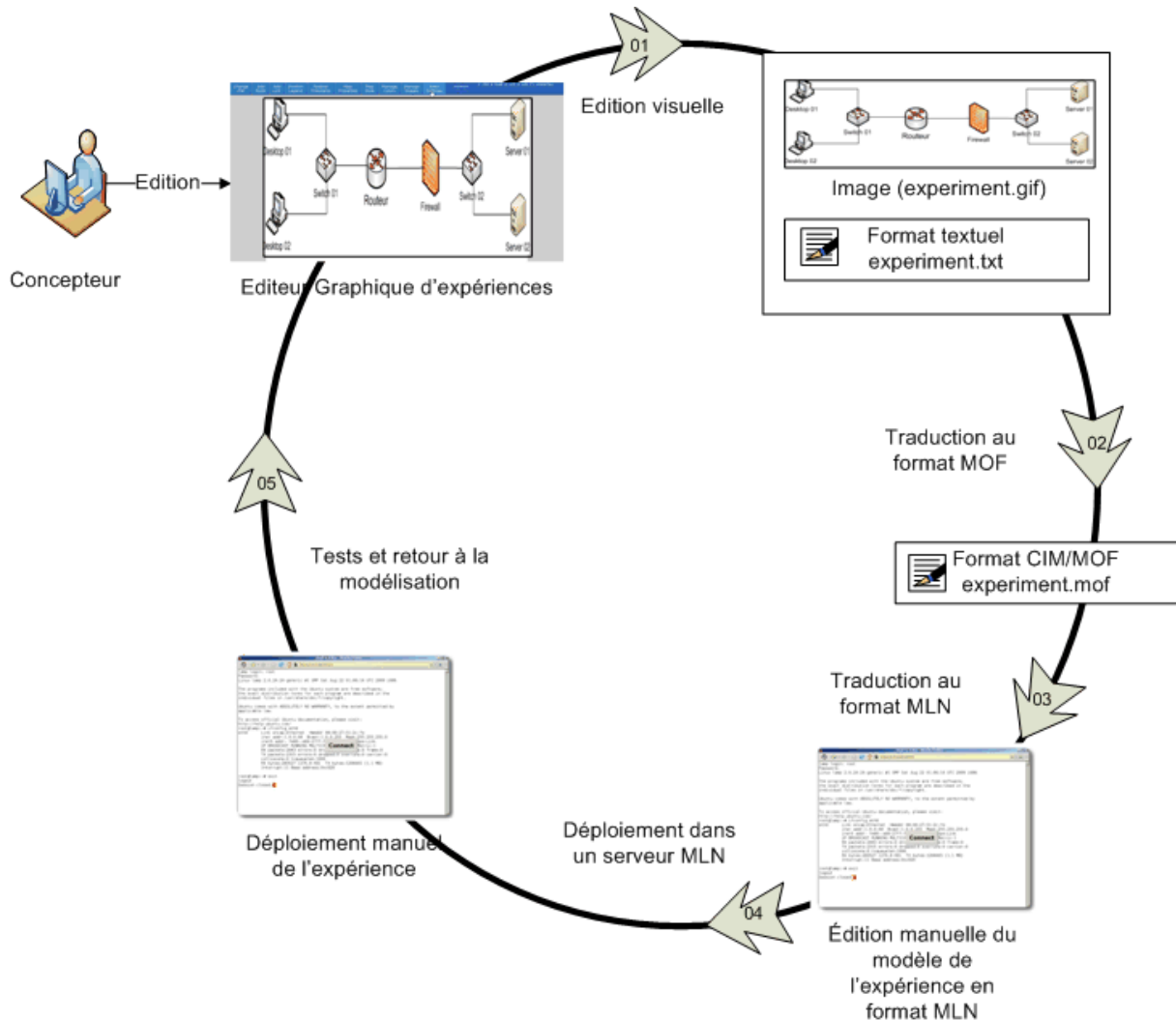


Figure 10-1. Processus de prototypage d'une expérience.

Dans la suite, nous expliquons en détail les outils et IHMs qui permettent de réaliser chacune de ces activités.

10.2.1. Conception des expériences avec un éditeur graphique

Dans un premier temps, nous nous sommes focalisés sur l'intégration et le développement d'une interface graphique d'édition d'expériences qui permet au concepteur d'explicitier sa vision de l'expérience. Cette IHM repose sur un outil existant qui répond partiellement à nos

besoins, l'outil Network-WeatherMap Editor⁸⁰. Son interface web constituée de scripts PHP et JavaScript facilement intégrable dans Moodle permet l'édition graphique d'un réseau, et produit un fichier de configuration décrivant l'expérience selon un langage textuel spécifique mais intuitif et bien documenté ; le modèle WeatherMap détaillé de notre expérience généré par notre version de l'éditeur est présenté en détail dans le Tableau E-3 de l'Annexe E. D'autre part, à partir du fichier produit et correspondant à l'expérience modélisée, l'outil génère l'image associée.

Les modifications que nous avons apportées à cet outil portent principalement sur l'enrichissement de son langage et de son interface de conception par des options de configuration proches des classes et attributs utilisés dans nos modèles CIM et MLN. La Figure 10-2 illustre l'écran de l'éditeur WeatherMap après l'édition de notre expérience.

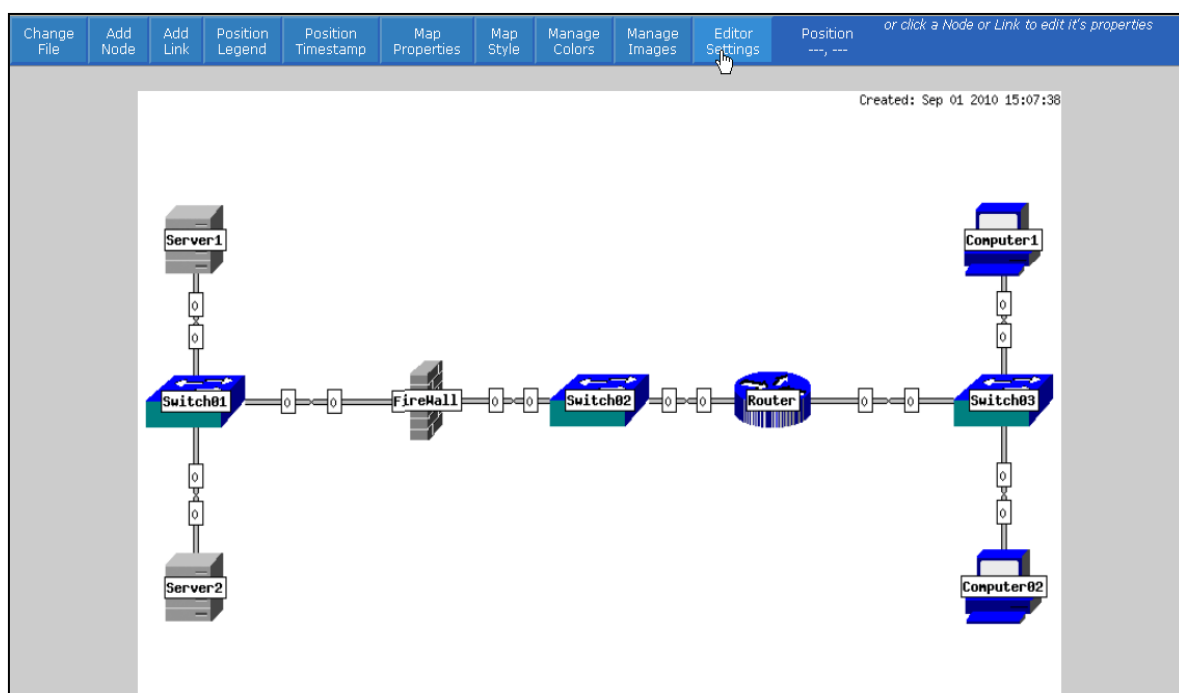


Figure 10-2. Conception de notre expérience réseau avec Network WeatherMap.

La barre de menus horizontale en haut de l'éditeur contient tous les boutons pour créer des nœuds et des liens entre eux, en plus de paramètres généraux. Par exemple, un lien est créé en cliquant sur le bouton *Add Link*, puis en sélectionnant successivement les deux nœuds

⁸⁰ <http://www.network-weathermap.com/>

correspondant aux extrémités du lien. Dans notre version de l'éditeur, le concepteur est invité à préciser l'interface réseau de chaque nœud qui va former l'extrémité du lien.

Un clic sur un nœud provoque l'ouverture d'une boîte de dialogue renseignant ses propriétés, que nous avons enrichie avec des champs spécifiques à nos modèles et détaillés dans le Tableau 10-1 ; la boîte de dialogue résultante apparaît sur la Figure 10-3.

Figure 10-3. Boite de dialogue des propriétés d'un nœud.

Tableau 10-1. Exemples d'options de configuration ajoutées à WeatherMap.

Option	Valeur	Exemple
os	Type et version du système d'exploitation.	SET os Linux:Ubuntu:9.04
ph-settings	Nombre de processeur, taille de la RAM et taille du disque.	SET ph-Settings:1:256M:2G
network	Type et nombre d'interfaces réseau.	SET network:ethernet:1
ip-settings	Paramètres IP d'une interface réseau (interface, type d'adressage, adresse IP, masque, passerelle, serveur DNS).	SET ip-settings eth0:static: 192.168.0.5:255.255.255.0:10.0.10.1:10.0.10.2

Une fois que l'enseignant concepteur a conçu son expérience à travers l'éditeur graphique, les phases suivantes pour la mise en œuvre d'une expérience correspondent à son instantiation dans le serveur WBEM, et à son déploiement sur la couche d'Expérimentation. Une étape

préliminaire consiste donc à assurer la correspondance entre le format WeatherMap et les formats MOF et MLN.

10.2.2. Traduction vers les formats MOF et MLN

Afin de simplifier au maximum la tâche de conception et de déploiement des activités de télé-TP, les opérations de traduction du modèle WeatherMap vers les formats MOF et MLN devraient être transparentes à l'utilisateur. Toutefois, dans l'implémentation actuelle, le concepteur doit utiliser une interface web additionnelle pour réaliser ces transformations. Lorsqu'il valide la conception de son expérience, l'enseignant est redirigé vers une fenêtre Terminal intégrée à l'IHM de Moodle afin d'exécuter successivement deux programmes en mode ligne de commande générant les fichiers MOF et MLN à partir du fichier de configuration WeatherMap.

10.2.2.1. Traduction du modèle de l'expérience du format visuel vers le format MOF

La première transformation vise la création d'un diagramme d'instances abstrait de l'expérience représentant l'objet pédagogique interactif « Expérience ». Cette phase couvre deux objectifs : superviser l'expérience, et indexer le modèle résultant dans un vivier d'objets pédagogiques à des fins de partage et de réutilisation. L'algorithme général de transformation du format WeatherMap vers le format MOF se déroule comme suit :

1. Générer une instance de la classe *TEL_ComputerExperiment* correspondant à l'objet pédagogique « Expérience ».
2. Générer, pour chaque enregistrement de type NODE (voir l'annexe E pour les détails), une instance de la classe *TEL_ComputerSystem*. Dans cette étape, il s'agit également de créer l'ensemble des instances correspondant aux composants internes de chaque nœud et d'établir les associations adéquates.
3. Générer une instance de la classe *LAB_TopologyGraph* ainsi que l'association *Component* la reliant à l'instance *LAB_ComputerExperiment*.
4. Pour modéliser la topologie du réseau, créer autant d'instances de la classe *LANConnectivitySegment* que de commutateurs puis, pour chacune d'entre elles, rajouter

une association de type *MemberOfCollection* avec l'instance de la classe *LAB_TopologyGraph*.

Une vérification syntaxique et sémantique est appliquée sur le modèle MOF produit par l'outil afin de déceler d'éventuelles erreurs et permettre leur correction manuelle. Une fois validé, le modèle est automatiquement injecté dans le serveur OpenPegasus pour être exploité lors de l'activité de télé-TP.

10.2.2.2. Traduction du modèle de l'expérience du format MOF vers le format MLN

A partir du modèle MOF généré, le concepteur peut procéder à sa transformation vers le format de la plateforme d'expérimentation qui correspond dans notre expérimentation à l'outil de virtualisation MLN et à la technologie XEN sous-jacente. Le fichier de configuration MLN produit par notre outil de conversion MOF/MLN contient les éléments suivants (consulter l'annexe E pour un exemple détaillé) :

1. La description du projet MLN qui correspond à la description générale de l'expérience (nom de l'expérience et paramètres généraux).
2. La spécification des différents commutateurs dédiés à l'interconnexion des nœuds de l'expérience, où chaque enregistrement MLN de type *Switch* correspond à une instance MOF de type *LANConnectivitySegment*.
3. Autant d'enregistrements de type "*host*" que d'instances MOF de type *TEL_ComputerExperiment*. Chaque enregistrement décrit le nombre de processeurs (en fonction du nombre d'instances *Processor* associées à l'instance *TEL_ComputerSystem*), la taille de la mémoire vive (en fonction de la valeur des attributs *BlockSize* et *NumberOfBlocks* de l'instance de la classe *Memory* associée à l'instance *TEL_ComputerSystem*), et le système de fichier à utiliser pour cette machine virtuelle (connu à partir de la combinaison des instances *FileSystem* et *OperatingSystem* associées à l'instance du nœud).
4. Autant d'enregistrements de type « network ethxx » que d'instances MOF de type *LANEndPoint*.

Une fois le modèle MLN généré, sa validité est vérifiée à travers l'option *parse* de la commande *mln* qui s'assure de la cohérence syntaxique et sémantique d'un fichier de

configuration donné. L'expérience peut ensuite être construite puis déployée en vue de l'activité de télé-TP.

Pour illustrer les algorithmes de traduction précédents, l'annexe E détaille une application concrète où le Tableau E-1 présente l'expérience en format WeatherMap, le tableau E-2 présente quelques instances CIM de l'expérience en format MOF, alors que le Tableau E-3 expose le modèle MLN associé à la même expérience type.

10.2.3. Gestion du cycle de vie des expériences

L'état actuel de l'IHM de conception ne permettant pas le déploiement de l'expérience, cette phase doit être effectuée soit par l'invocation des méthodes de gestion du cycle de vie de la classe *LAB_ComputerExperiment* (en utilisant un client WBEM intégré à la plateforme d'apprentissage Moodle), soit par l'accès direct à la plateforme MLN et l'invocation de commandes spécifiques à cet outil. La première méthode doit être privilégiée car elle permet d'exploiter (et tester) nos différentes contributions qui visent à manipuler les expériences d'une manière indépendante de la plateforme d'expérimentation. Toutefois, la deuxième méthode peut être utile lorsque certaines méthodes ne sont pas implémentées, ou pour effectuer des vérifications directement sur la plateforme.

La suite de cette section établit la correspondance des différentes opérations du cycle de vie d'une expérience avec les méthodes WBEM développées, et expose les commandes MLN associées pour plus de clarté. Notons que l'interaction avec le serveur OpenPegasus est assurée à travers un client WBEM livré avec cet outil, et en particulier avec la commande *cimcli*. Cette commande a également été exécutée pour le développement de notre moteur d'exécution des expériences, puisque les scripts PHP et CGI qui composent les agents du moteur l'ont utilisé de manière intensive.

10.2.3.1. Construction des machines virtuelles de l'expérience

Avant de déployer l'expérience réseau conçue par l'enseignant sur l'outil MLN, il faut s'assurer que les nœuds qui la constituent sont correctement construits : il s'agit donc de construire les machines virtuelles nécessaires à la mise en œuvre de l'expérience et qui suivent le même cycle de vie que cette dernière (pour alléger la lecture de ce chapitre, nous avons renvoyé les détails de ce cycle dans l'Annexe D).

Généralement, une expérience réseau utilise une bibliothèque de machines virtuelles « modèles » qui vont servir de base pour la construction des machines virtuelles réellement utilisées par l'expérience déployée. Une machine virtuelle « modèle » est un fichier image contenant un système d'exploitation spécifique (comme Linux) reposant sur un système de fichier adéquat (comme EXT3), doté d'un ensemble de logiciels préinstallés (routeur, pare-feu, services, applications de bureau, etc.) et caractérisé par une configuration initiale spécifique. Construites ainsi par un concepteur ou un expert tiers, les machines modèles sont destinées à être stockées et partagées dans des viviers de machines virtuelles afin d'épargner à la communauté le même travail de conception. Différents outils permettent de concevoir, construire et partager des bibliothèques de machines virtuelles modèles ; dans notre implémentation, nous avons sélectionné le système RBuilder⁸¹ qui supporte la majorité des systèmes de virtualisation mais qui, surtout, offre la possibilité de concevoir finement des machines virtuelles. La particularité de nos machines virtuelles modèles réside dans le fait qu'elles sont dotées du ou des Pilotes WBEM autonomes que nous avons développés, et qui sont nécessaires au contrôle et à la supervision des machines virtuelles par le serveur OpenPegasus.

Dans le cas de notre scénario type, cinq nœuds illustrés par la Figure 10-4 constituent l'expérience (les commutateurs étant virtualisés par l'outil MLN sous la forme d'un programme) : *Serveur01*, *Server02*, *Desktop01*, *Desktop02*, *Firewall* et *Router*. Pour construire les machines virtuelles correspondantes, l'enseignant concepteur doit invoquer, via le programme *cimcli*, la méthode *Create()* de l'objet *LAB_ComputerExperiment* :

cimcli -l serveur-WBEM im "LAB_ComputerExperiment.ExperimentID=NetExp01" Create

L'invocation de cette méthode sur le serveur WBEM va en fait engendrer une interaction avec le Pilote WBEM de l'outil MLN sur la plateforme d'expérimentation, qui va exécuter la commande « *mln buid -f* » suivie du fichier contenant la configuration du projet. Le programme génère alors les instances des systèmes de fichier de chaque machine virtuelle composant l'expérience.

⁸¹ [Http://www.rpath.com](http://www.rpath.com)

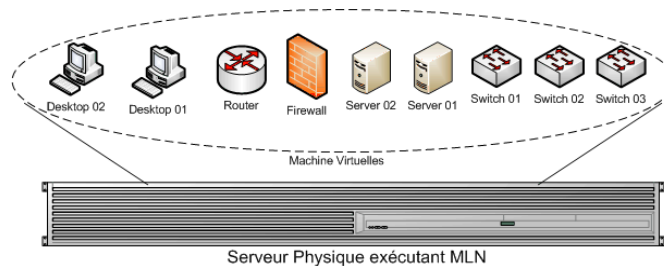


Figure 10-2. Machines virtuelles nécessaires à notre expérience.

Précisons qu’à ce niveau, les machines virtuelles sont construites seulement et ne sont pas encore opérationnelles. Elles le seront lors de la phase de déploiement de l’expérience décrite dans la prochaine section.

10.2.3.2. Déploiement des expériences

La phase de déploiement d’une expérience réseau consiste à exécuter les machines virtuelles construites lors de la précédente phase en les dotant des ressources spécifiées dans le modèle de l’expérience (processeurs, mémoire vive, ports réseaux virtuels, etc.). Durant une phase préliminaire, MLN déploie les commutateurs virtuels qui vont interconnecter les ports des machines virtuelles selon la topologie du réseau spécifique à l’expérience (voir Figure 10-5).

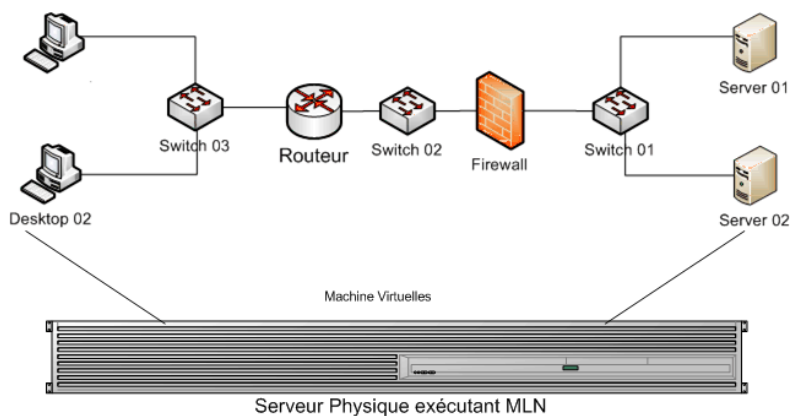


Figure 10-3. Mise en place de la topologie de notre expérience.

Une fois les machines virtuelles opérationnelles, l’expérience peut être déployée sur demande du concepteur à travers la commande suivante :

cimcli -l serveur-WBEM im “LAB_ComputerExperiment.ExperimentId=NetExp01” Start

La requête WBEM ci-dessus est transmise au serveur OpenPegasus qui invoque le Pilote WBEM de MLN pour exécuter la commande “*mln start -p NetExp01*” et assurer le

déploiement effectif de l'expérience. Les Pilotes WBEM embarqués dans chaque machine virtuelle sont alors actifs et commencent à communiquer avec le serveur OpenPegasus ; une description détaillée des Pilotes et des classes qu'ils implémentent est donnée dans la section 9.6.3.

Souvent une succession d'allers-retours entre les étapes de modélisation, de construction et de déploiement précède la validation du modèle définitif de l'expérience. A partir du déploiement de l'expérience, le concepteur est en mesure de vérifier son bon fonctionnement et y apporter d'éventuels corrections et affinements ; il dispose pour cela de l'interface d'exploitation offerte aux enseignants tuteurs que nous présentons plus bas.

10.2.3.3. Autres phases du cycle de vie d'une expérience réseau

Les autres phases du cycle de vie d'une expérience réseau sont implémentées sous la forme de différentes méthodes de la classe *LAB_ComputerExperiment* présentées dans le Tableau 10-2.

Tableau 10-2. Méthodes implémentées pour la gestion du cycle de vie d'une expérience réseau.

Phase	Méthode	Commande WBEM	Commande MLN	Commentaire
Arrêt	<i>Stop</i>	<i>cimcli -l serveur-wbem im "LAB_CompExp.ID =NetExp01" Stop</i>	<i>mln stop -p nom-expérience</i>	Arrêt des commutateurs virtuels et des machines virtuelles avec la libération des ressources utilisées par chacune de ces entités (espace mémoire, CPU, etc.)
Suppression	<i>Remove</i>	<i>cimcli -l serveur-wbem im "LAB_CompExp.ID =NetExp01" Remove</i>	<i>mln remove -p nom-expérience</i>	Suppression des machines et commutateurs virtuels qui composent l'expérience.
Sauvegarde	<i>Save</i>	<i>cimcli -l serveur-wbem im "LAB_CompExp.ID</i>	<i>mln export -p nom-expérience</i>	Stockage de l'expérience dans son

		<code>=NetExp01" Save</code>	<code>-d repertoire- stockage</code>	état actuel au sein d'un répertoire dédié.
Restauration	<i>Restore</i>	<code>cimcli -l serveur-wbem im "LAB_CompExp.ID =NetExp01" Restore</code>	<code>mln import -p nom-expérience -d repertoire- stockage</code>	Copie d'un fichier de sauvegarde de l'expérience dans le répertoire de travail de MLN.

Outre la conception et le déploiement d'une expérience, les enseignants sont encouragés à coopérer et collaborer entre eux en utilisant un vivier d'objets pédagogiques « Expérience » conçus, testés et partagés afin d'être agrégés ou réutilisés. Ce vivier n'est pas forcément situé au sein du LMS, et il est même recommandé de l'installer dans un système dédié pour favoriser le partage des objets pédagogiques à une plus grande échelle.

10.2.4. Stockage et partage des objets pédagogiques « Expérience »

Nous rappelons ici que les objets pédagogiques de type « Expérience » sont mis à la disposition de la communauté grâce aux viviers d'objets pédagogiques afin de favoriser le partage et la mutualisation de ce nouveau type de ressources. Ces bases de données d'objets pédagogiques contiennent les fiches descriptives conformes au standard LOM (notre proposition de profil LOM pour les objets « Expérience » a déjà été présentée dans la section 5.2 et illustrée par la Figure 5-3), et indiquent la localisation du modèle de l'expérience au format XML.

Notre équipe de recherche étant fortement impliquée dans la fondation ARIADNE depuis plus de dix ans, la fiche LOM de notre expérience réseau est stockée dans un vivier de connaissances fondé sur les technologies ARIADNE et local à notre université. En particulier, le service d'indexation *Simple Publishing Interface* (SPI) est invoqué pour assurer cette tâche.

10.3. Outils et IHMs supports aux activités d'apprentissage individuel

Dans cette section nous présentons l'IHM mise à disposition des apprenants qui supporte l'activité d'apprentissage individuel. Cette interface repose sur la technologie AJAX dont la

partie cliente comprend un ensemble de scripts JavaScript intégrant la librairie JQUERY, alors que la partie serveur développée en PHP et en CGI implémente le moteur d'exécution des expériences qui s'occupe de faire le lien entre l'interface et la plateforme Moodle. Afin de maintenir une interface unique et légère, nous avons opté pour trois types de contrôles familiers aux utilisateurs : (1) le contrôle de type arborescence pour explorer de façon hiérarchique des objets complexes comme les expériences ou la liste des utilisateurs, (2) les onglets pour regrouper dans un même espace plusieurs fenêtres de même type, et (3) les menus contextuels pour associer à chaque élément de l'espace de travail des opérations spécifiques et éviter des barres de menus complexes.

L'interface est une implémentation de l'IHM générique de télé-opération présentée dans la section 5.2.3 fondée sur la métaphore du « Cockpit », et augmentée d'outils de communication et de collaboration. Notre IHM d'apprentissage est composée de cinq régions redimensionnables illustrées par la Figure 10-6 :

- Une barre de menu offre des fonctions communes à toute l'expérience et permet de personnaliser l'affichage. Les contrôles implémentés actuellement concernent la déconnexion à la plateforme et le masquage/affichage des trois régions latérales droite, gauche et basse. Les autres contrôles prévus devraient assurer la sauvegarde de l'expérience en cours et le passage en mode édition en vue d'éventuelles modifications sur l'expérience.
- Un explorateur de l'expérience qui utilise un contrôle de type arborescence.
- La région centrale exploite les onglets pour regrouper toutes les fenêtres de sessions Terminal ouvertes sur les machines de l'expérience.
- Un explorateur des utilisateurs liste les apprenants, tuteurs ou enseignants associés à l'expérience, et indique leur état de présence ou d'absence. Il offre également d'autres fonctionnalités dédiées à l'apprentissage collaboratif.
- Un outil de messagerie instantanée supporte la communication et la collaboration entre les acteurs durant l'expérimentation.

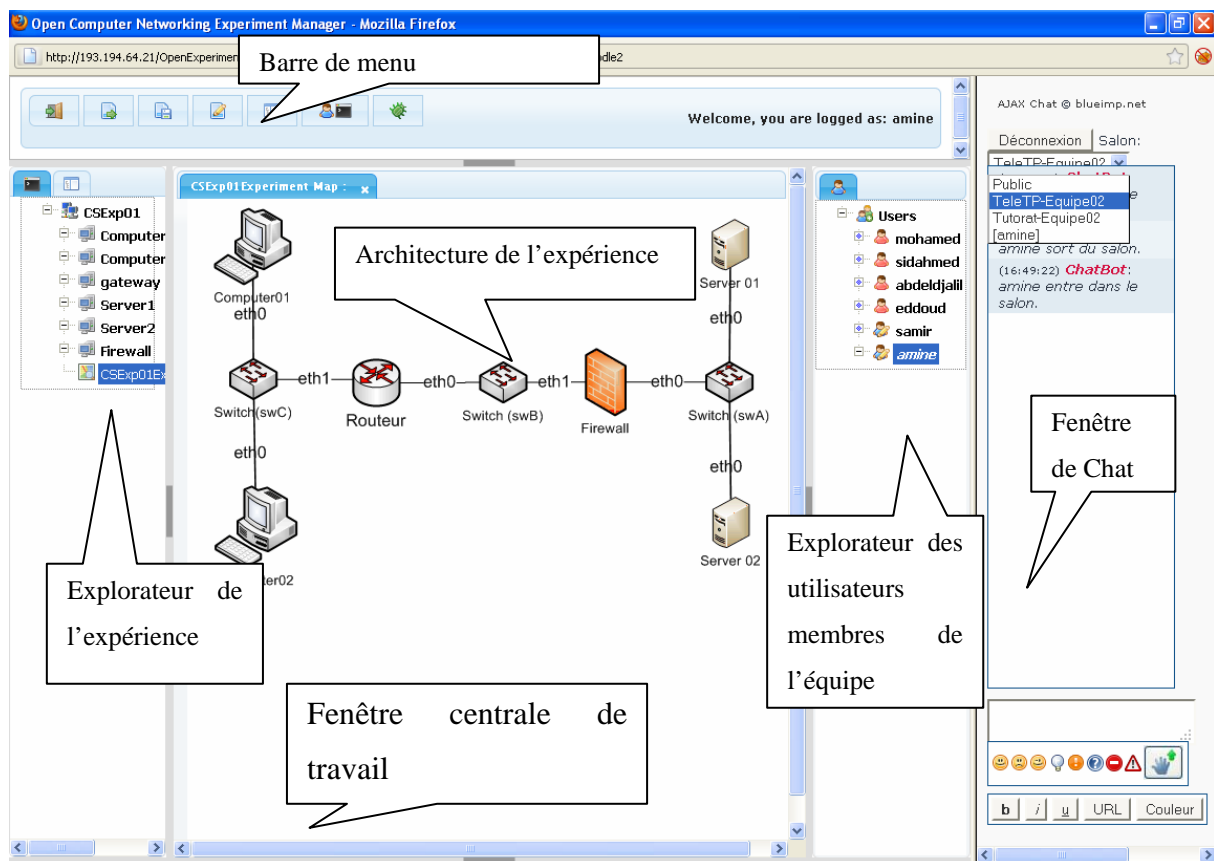


Figure 10-6. Une IHM pour le support des activités d'apprentissage.

Nous détaillons dans cette section les trois premiers points, les deux dernières fonctionnalités se focalisant sur les activités d'apprentissage collaboratif.

10.3.1. La barre de menus

La barre de menus illustrée par la Figure 10-7 est composée d'un groupe de sept boutons suivi d'un message de bienvenu indiquant l'identité de l'utilisateur connecté sur l'application. La connexion d'un utilisateur sur notre plateforme est liée à la présence dans le serveur OpenPegasus d'une instance de la classe *CIM_Identity* qui représente cet utilisateur ; son état de connexion se traduit par le positionnement de l'attribut booléen *CurrentlyAuthenticated* à *True* pour exprimer le fait qu'il est actuellement connecté, ou *False* sinon.



Figure 10-4. Détails de la barre de menus de l'IHM d'exploitation d'une expérience réseau.

Les fonctions assurées par ces boutons sont respectivement (par ordre d'apparence sur la figure ci-dessus) :

1. La déconnexion à l'application qui implique dans le serveur OpenPegasus la mise à jour de la valeur de l'attribut *CurrentlyAuthenticated* à *False*.
2. Le lancement d'une expérience provoque l'ouverture d'une boîte de dialogue proposant une liste d'expériences disponibles. Lorsqu'une expérience est sélectionnée, une requête est envoyée à travers tout le système à la plateforme de tests afin de démarrer cette expérience. Cette fonctionnalité n'est pas encore implémentée.
3. L'enregistrement de l'expérience dans son état actuel se traduit par l'enregistrement des instances du modèle CIM de l'expérience avec les valeurs actuelles des attributs. D'autre part, l'opération de stockage de l'expérience sur le système MLN est exécutée selon le processus décrit dans la section précédente. Cette fonctionnalité n'est pas encore implémentée.
4. Le passage en mode édition de l'expérience en réutilisant l'interface de conception décrite précédemment. Cette fonction n'est disponible que pour les enseignants concepteurs et les utilisateurs autorisés par ces derniers. Cette fonctionnalité n'est pas encore implémentée.
5. Cacher/Montrer la barre gauche dédiée à l'exploration de l'expérience.
6. Cacher/Montrer la barre droite dédiée à l'exploration des utilisateurs et de leurs sessions.
7. Cacher/Montrer la barre basse dédiée à l'affichage des différents types de messages d'erreurs, d'avertissement ou de débogage.

10.3.2. L'explorateur des expériences

L'explorateur des expériences présenté dans la partie gauche de la Figure 10-8 permet de visualiser, de manière hiérarchisée, l'ensemble des expériences auxquelles participe un apprenant ainsi que les nœuds qui la composent ; il constitue donc le point d'entrée à la création et à l'exploitation de sessions Terminal. Pour offrir ces fonctionnalités, l'explorateur retrouve, via les services de la couche d'Intégration, les informations contenues dans le modèle de l'expérience instancié au sein du serveur OpenPegasus et illustré par la partie droite de la Figure 10-6. L'application explore tout d'abord, grâce à l'association *TEL_IdentityOnResource*, les expériences associées à l'utilisateur courant représenté dans le système par une instance de type *CIM_Identity*. Une fois les instances des expériences

retrouvées, l'application cherche tous les nœuds (de type *TEL_ComputerSystem*) associés à une expérience en retrouvant les associations *Component*.

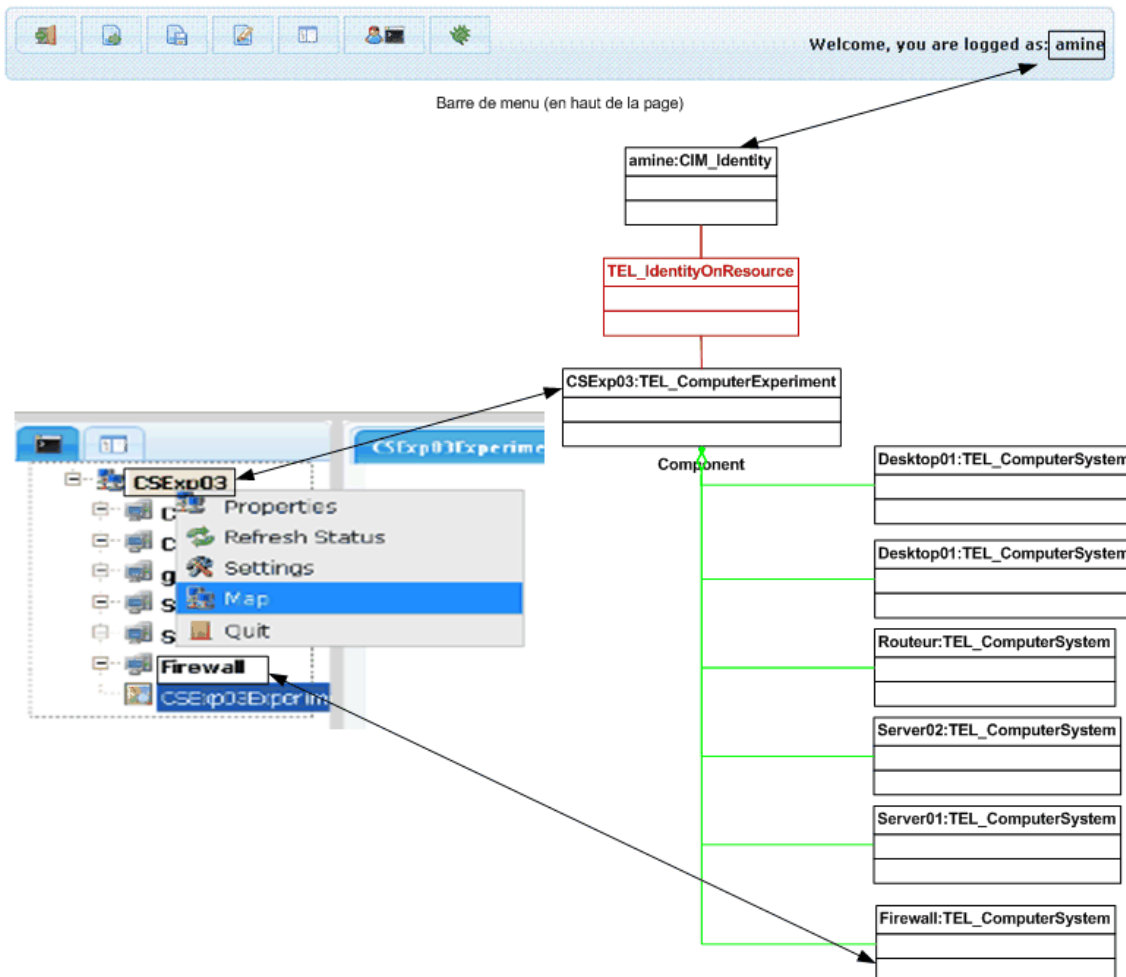


Figure 10-8. L'explorateur de l'expérience.

En cliquant du bouton droit de la souris sur le nom de l'expérience (qui est la racine de l'arborescence de l'expérience), un menu contextuel contenant un ensemble d'options est affiché. Il est possible grâce à ce menu d'obtenir des informations générales (description générale, concepteur, date de création, etc.) sur l'expérience (option *Properties*), d'en rafraîchir l'état (option *Refresh Status*), d'obtenir et afficher les valeurs des attributs de l'instance *TEL_ComputerExperiment* (option *Settings*), et d'en afficher la topologie (option *Map*). Enfin, l'option *Quit* masque ce menu contextuel.

10.3.2.1. Contrôle et supervision des nœuds de l'expérience

Le contrôle et la supervision des nœuds de l'expérience s'effectuent à travers les options du menu contextuel associé à ceux-ci (voir Figure 10-9) qui permettent :

- D'afficher une description générale de la machine (option *Properties*).
- D'afficher les propriétés générales de la machine à partir des attributs de l'instance *TEL_ComputerSystem* (option *Settings*).
- De rafraîchir l'état de la machine dans l'explorateur (option *Refresh*).
- D'arrêter la machine (option *Stop*) en invoquant la méthode CIM native *SetPowerState* ("Power Off") sur l'instance *TEL_ComputerSystem*.
- De démarrer la machine (option *Start*) en invoquant la méthode native *SetPowerState* ("Full Power") sur l'instance *TEL_ComputerSystem*.
- De redémarrer la machine (option *Reboot*) en invoquant successivement les deux méthodes natives des deux options précédentes.
- De réinitialiser la machine avec ses propriétés d'origine (option *Reset*).

Ainsi ces capacités favorisent l'apprentissage par essai/erreur et l'extension de la durée du télé-TP, en plus de fournir des moyens simples et intuitifs pour réaliser des opérations complexes.

10.3.2.2. Exploration et exploitation des sessions en cours

Pour effectuer des opérations spécifiques sur une machine particulière, l'utilisateur doit ouvrir une session Terminal sur celle-ci selon deux types de session distincts présentés dans le chapitre précédent : les terminaux console et les terminaux réseau (options *Console Access* et *SSH Access* respectivement). Une fois le mode de session déterminé, une boîte de dialogue illustrée dans la partie droite de la Figure 10-9 demande à l'utilisateur de choisir les privilèges d'accès des autres utilisateurs sur sa session (aucun, lecture, lecture et écriture) ; par défaut la session Terminal est privée. Une session accessible en lecture va permettre à d'autres utilisateurs de la rejouer ou de la regarder en temps réel, alors qu'une session accessible en lecture et écriture (session partagée) va leur permettre, en plus, d'accéder en contrôle sur cette session.

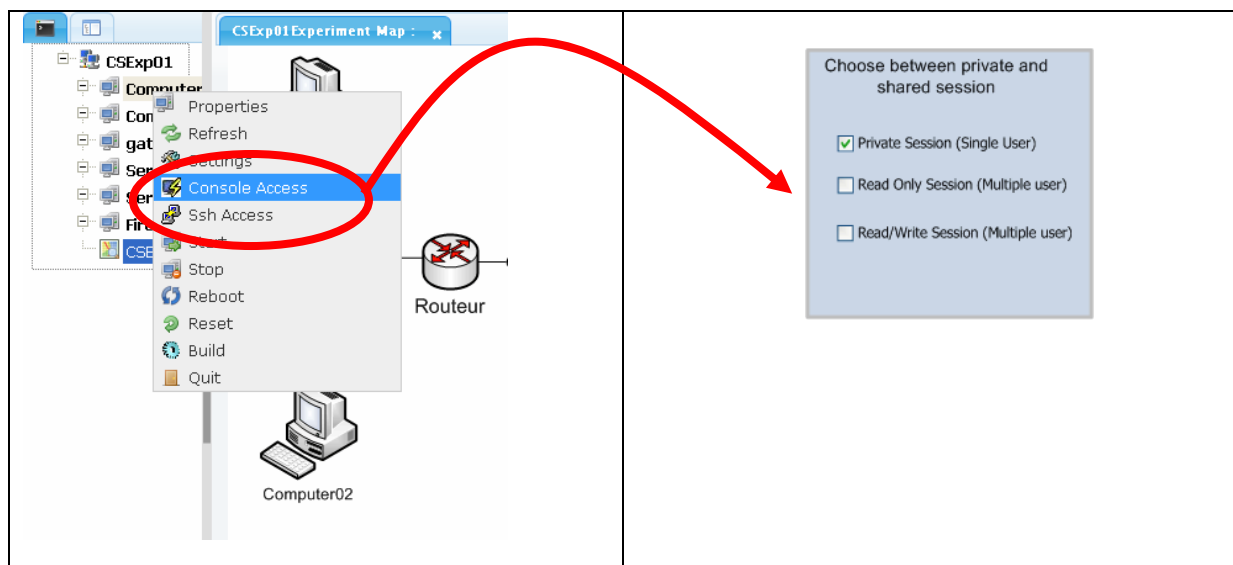


Figure 10-9. Menu contextuel associé à un nœud de l'expérience.

L'ouverture d'une nouvelle session implique la génération des instances des classes et associations liées au modèle d'activités des acteurs humains sur l'expérience (voir la section 9.3.4 et l'exemple donné dans la Figure 9-15). En particulier, une nouvelle instance de la classe *TEL_TTYSession* est créée, dont les valeurs de ses attributs sont déduites à partir des autres informations renfermées dans le serveur OpenPegasus. Rappelons ici que chaque commande émise par un utilisateur (ainsi que les réactions de la machine virtuelle) est enregistrée par le programme *WBEM_TTYREC* présenté dans le chapitre 9 et qui s'exécute au niveau du serveur OpenPegasus. Ainsi chaque trace d'activité est enregistrée dans un objet CIM de type *TEL_CommandExperimentActivity* et pour chaque instance de ce type, une association *TEL_ActivityOnResourceComponent* est établie avec l'instance CIM représentant la session. A travers ce mécanisme, une session peut être fidèlement reconstituée et rejouée a posteriori comme elle peut être visualisée en temps réel.

En plus du menu contextuel associé à chaque nœud, le niveau inférieur de l'arborescence de l'explorateur de l'expérience permet de visualiser les sessions ouvertes sur chacun des nœuds, comme l'illustre la Figure 10-10. Chaque session est rattachée à la machine sur laquelle elle a été lancée, et l'icône associée en indique le type. Ces fonctionnalités sont rendues possibles grâce aux instances de type *TEL_TTYSession* qui sont retrouvées à travers les associations *TEL_DependencyResource* avec les instances des nœuds de l'expérience, elles-mêmes retrouvées via l'association *TEL_IdentityOnResource* avec l'utilisateur courant.

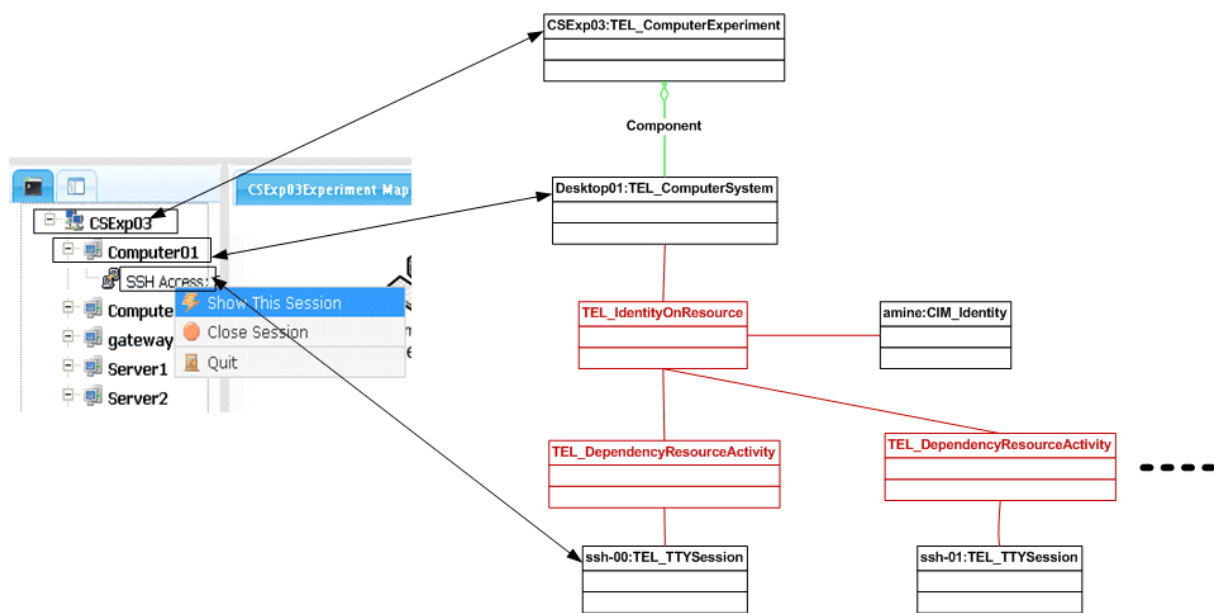


Figure 10-10. Menu contextuel associé à une session ouverte sur un nœud de l'expérience.

Enfin, l'interface est conçue de telle sorte que la sélection d'une session dans l'explorateur de l'expérience va activer l'élément correspondant dans la région centrale, et contribue à aider l'apprenant à repérer facilement ses sessions ; il peut réaliser la même opération à travers l'option « *Show This Session* » du menu contextuel associée aux éléments de type session et illustrée par la Figure 10-10. En outre, l'option « *Close Session* » permet de clôturer définitivement la session, ce qui se traduit d'un côté par l'arrêt des processus Shellinbox et WBEM_TTYREC ainsi que de la session console ou de la connexion SSH, et d'un autre côté par la mise à jour de l'attribut *EnabledState* de l'instance *TEL_TTYSession* avec la valeur 3 (*Inactif*).

10.3.3. Exploitation de l'expérience

Nous l'avons mentionné précédemment, l'exploitation d'une expérience consiste à ouvrir des sessions Terminal sur les différents nœuds de celle-ci. Notre IHM intègre donc l'outil Shellinbox qui se présente sous la forme d'une interface web reproduisant l'apparence et le comportement d'un Terminal réel. Cette fidélité technique favorise l'atteinte des objectifs pédagogiques particuliers de maîtrise des compétences techniques et professionnelles, et augmente la motivation des apprenants qui ont l'impression de manipuler un système réel (voir Figure 10-11).

10.3.3.1. Navigation et accès aux sessions ouvertes

Quels que soient le type de session et le mode d'accès sélectionnés par un apprenant lors de l'ouverture d'une nouvelle session sur un nœud de l'expérience, l'onglet d'une fenêtre Terminal va s'ouvrir dans la région centrale de l'espace de travail afin qu'il puisse y exécuter des commandes et programmes. Cette zone, illustrée par la Figure 10-11, regroupe donc toutes les fenêtres des sessions ouvertes par l'apprenant, sur la même machine ou sur des nœuds différents, afin de faciliter la navigation parmi celles-ci. Ces contrôles de type Onglets sont couplés à l'explorateur des expériences afin que la sélection de l'onglet d'une session déclenche automatiquement la mise en évidence de l'élément de l'expérience relatif à cette même session (et vice-versa). Ainsi l'utilisateur dispose d'un contrôle total sur les sessions qu'il a ouvertes grâce à l'affichage dans la barre gauche des éléments de type session (attachés aux nœuds de l'expérience sur lesquelles les sessions ont été ouvertes), et à la liste des onglets correspondants à ces sessions au centre de l'interface.

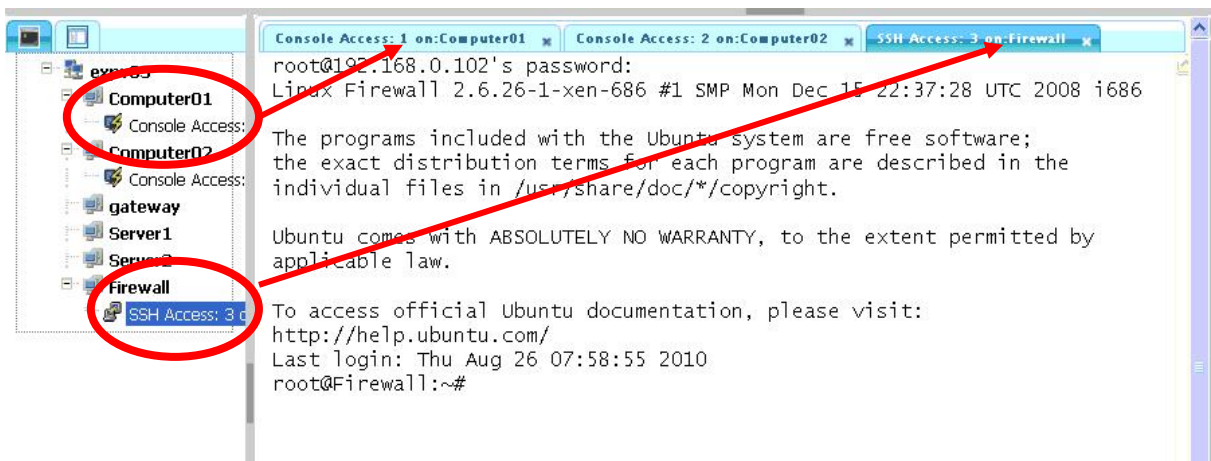


Figure 10-11. Navigation entre différentes sessions Terminal.

10.3.3.2. Reprise de sessions persistantes

Pour diverses raisons, un utilisateur peut être amené à se déconnecter de l'application en abandonnant des sessions ouvertes, et à les reprendre lors d'une autre séance de travail par exemple. Lorsqu'un apprenant se reconnecte à l'expérience, le système recherche et restaure toutes ses sessions actives tout en reconstruisant l'arborescence de l'explorateur de l'expérience, comme le montre la Figure 10-12. Cette capacité est possible grâce à l'attribut *EnabledState* de la classe *TEL_TTYSession* qui indique que la session est active ou non.

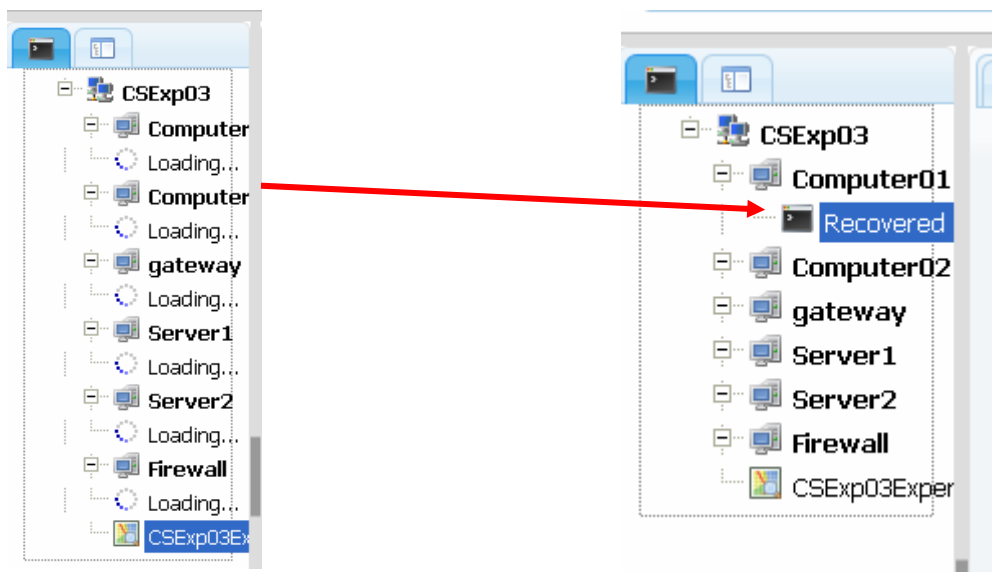


Figure 10-12. Recherche d'éventuelles sessions existantes et restauration d'une session retrouvée.

Cette fonctionnalité présente deux atouts puisqu'elle diminue le nombre de sessions actives par machines d'une part, et représente un moyen de reprise d'une session existante après une déconnexion volontaire ou involontaire d'autre part ; nous verrons plus loin les alternatives offrant cette dernière fonctionnalité.

10.3.3.3. Recommandations relatives aux sessions

La capacité d'ouvrir différentes sessions sur la même machine est essentielle lorsqu'il s'agit d'exécuter des tâches sur une fenêtre et consulter leurs résultats en temps réel dans une autre fenêtre. Bien que notre espace de travail supporte un nombre illimité de sessions pour chaque nœud d'une expérience, cette facilité doit être utilisée avec précaution.

En effet, il n'est pas conseillé d'ouvrir plusieurs sessions Terminal de type Console sur la même machine, au risque de bloquer les autres ou soi-même. Par ailleurs, même si de nombreuses sessions SSH peuvent être actives sur une même machine, l'abus de cette fonctionnalité pourrait mener au morcellement des traces d'activités et rendre difficile aux enseignants et apprenants la tâche d'analyse de la logique des opérations menées. Les tuteurs devraient encourager les apprenants à utiliser un nombre réduit de sessions et profiter de la persistance des sessions tout au long du télé-TP.

10.3.4. L'explorateur des utilisateurs

L'explorateur des utilisateurs, situé dans la partie droite de notre IHM, retrouve les utilisateurs participant aux expériences (voir la partie gauche de la Figure 10-13), mais offre avant tout à l'apprenant un moyen de gérer ses propres sessions et de consulter ses propres traces. Ainsi, lorsqu'un utilisateur est connecté, l'explorateur recherche toutes ses sessions Terminal actives ou terminées sur le serveur OpenPegasus, et les affiche dans l'interface de manière arborescente (voir la partie gauche de la Figure 10-13) ; l'icône associée à chaque session en indique le type (Console, SSH ou Terminée).

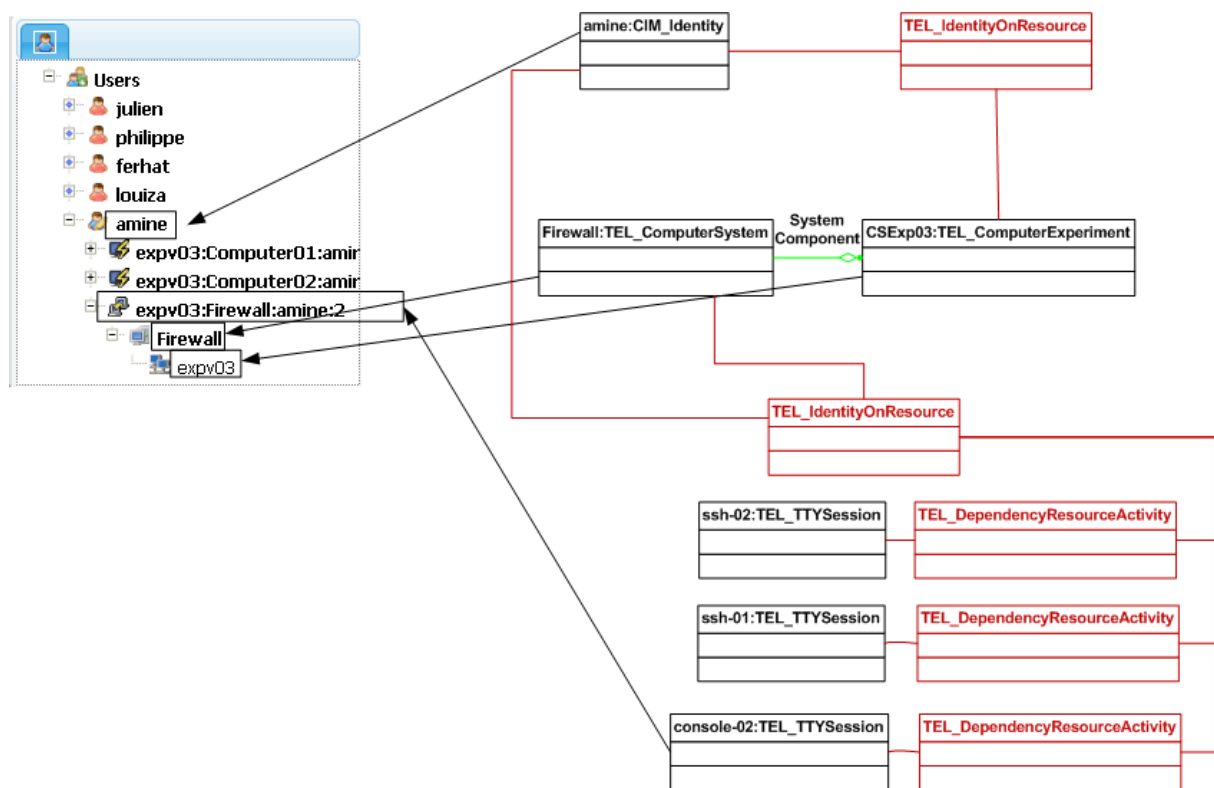


Figure 10-13. Structure de l'explorateur des utilisateurs et correspondance des éléments affichés avec les instances du modèle d'activités.

La partie droite de la Figure 10-13 illustre la correspondance entre les éléments affichés par cet explorateur et les instances du modèle d'activités au niveau du serveur OpenPegasus. Pour construire la liste des utilisateurs, l'application recherche, à partir de l'instance *TEL_ComputerExperiment*, toutes les instances *CIM_Identity* qui ont été associées à cette expérience à travers une instance *TEL_IdentityOnResource*.

La Figure 10-14b expose le menu contextuel associé à une session terminée. Ce menu ne contient que l'option *Replay Session* qui permet de rejouer le déroulement de la session Terminal jusqu'à la dernière commande entrée par l'apprenant. Dans le cas d'une session active, le menu contextuel de la Figure 10-14c propose deux options supplémentaires : *View Session* permet de visualiser en temps réel le déroulement d'une session, alors que *Join Session* fournit à l'apprenant un moyen de reprendre sa session (voir section 10.3.3).

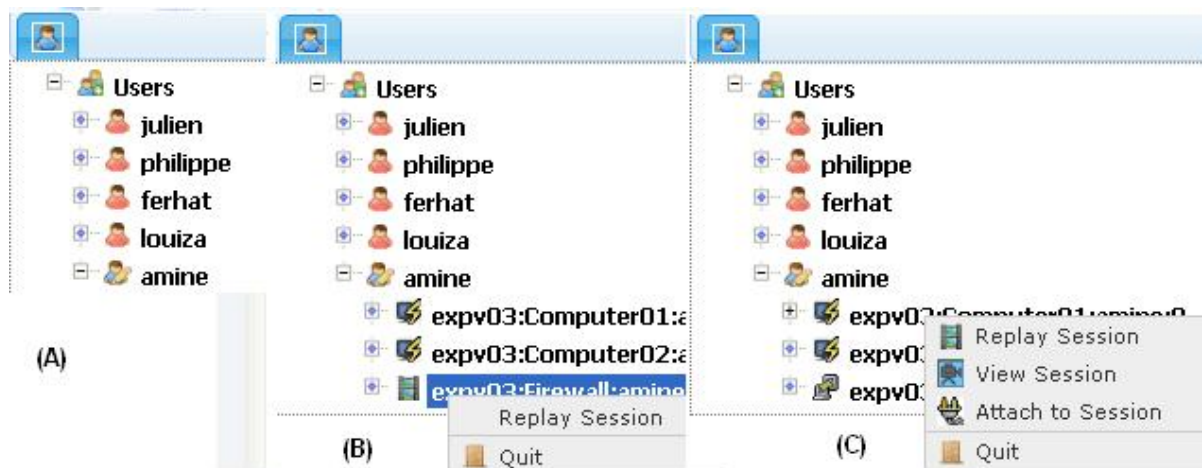


Figure 10-14. Exploration de ses propres sessions à partir de l'explorateur des utilisateurs.

La reprise d'une session active ou la consultation d'une session terminée aura comme conséquence l'ouverture de la fenêtre correspondante dans la région centrale (prise de contrôle sur la session dans le premier cas, ou simple visualisation dans le second). D'autre part, l'arborescence de l'explorateur de l'expérience va étendre les nœuds auxquels les sessions sont associées.

Notons enfin que l'explorateur des sessions des utilisateurs utilise les mêmes icônes pour symboliser les sessions, mais une icône supplémentaire symbolise les sessions terminées et archivées sous la forme d'un cliché pour indiquer qu'il n'est possible que de la consulter. Ces différentes fonctions permettent à l'apprenant de gérer son apprentissage en consultant les activités qu'il a effectuées, ou en reprenant des sessions qu'il avait perdues ou qu'il avait laissées délibérément actives.

10.4. Description de l'espace d'apprentissage collaboratif

10.4.1. Le sentiment de présence

Comme mentionné dans la section précédente, l'explorateur des utilisateurs a pour objectif l'identification de la liste des acteurs humains (apprenants et enseignants) impliqués dans la ou les mêmes expériences que celles de l'utilisateur connecté, ainsi que l'affichage de leur état de connexion. Cette liste d'utilisateurs est obtenue à partir des instances de la classe *CIM_Identity* qui sont liées, via l'association *TEL_IdentityOnResource*, à (aux) l'expérience(s) à (aux) laquelle(s) l'utilisateur connecté est rattaché. L'état de connexion d'un utilisateur correspond à son état de connexion dans Moodle, repris et indiqué au sein des instances de la classe *CIM_Identity* du serveur OpenPegasus (précisément grâce à l'attribut *CurrentlyAuthenticated*).

Un apprenant peut donc facilement identifier les divers acteurs impliqués dans l'expérience au même moment que lui grâce à la couleur de l'icône associée à chaque utilisateur (verte pour les utilisateurs connectés, rouge sinon), et ainsi bénéficier des atouts du sentiment de présence sociale. Nous avons déjà mentionné à plusieurs reprises dans la partie I de ce manuscrit qu'il s'agit d'un facteur renforçant la motivation des apprenants, et par conséquent l'efficacité pédagogique des activités de télé-TP. La présence sociale est également un élément important pour améliorer le travail collaboratif et le tutorat en ligne car il va obliger les acteurs à coordonner leurs activités pour atteindre les objectifs de l'activité.

D'autre part, en développant l'élément de l'IHM représentant un utilisateur, l'explorateur retrouve la liste de ses sessions afin qu'elles soient exploitées par d'autres acteurs de l'activité de télé-TP. Les outils offerts pour l'exploitation des sessions des autres utilisateurs, que nous présentons dans les prochaines sections, sont conçus de sorte à supporter d'une manière efficace le travail d'équipe et le suivi en ligne.

10.4.2. Exploitation de sessions ouvertes par d'autres acteurs

Si un utilisateur a la possibilité de rejouer ses propres sessions Terminal terminées ou de reprendre ses sessions actives, il a aussi l'opportunité d'effectuer les mêmes opérations avec des sessions ouvertes par d'autres utilisateurs (apprenants et enseignants). Ces fonctionnalités

dépendent du mode d'ouverture d'une session, puisque celle-ci doit être accessible en lecture. Dans ce cas, il suffit de sélectionner l'utilisateur ciblé, d'identifier et de cliquer du bouton droit sur la session désirée, puis de choisir l'option « *Replay Session* » pour rejouer une session terminée ou « *View Session* » pour visualiser en temps réel les commandes opérées par l'utilisateur dans une session active. Les processus d'identification et d'ouverture des sessions créées par d'autres utilisateurs sont les mêmes que ceux permettant d'explorer et d'exploiter ses propres sessions (voir la section 10.3.4). La seule différence réside dans le contrôle opéré par le système qui vérifie les droits d'accès à ces sessions à travers l'attribut *ReadWrite* de la classe *TEL_TTYSession*.

Ces fonctions permettent de supporter différentes activités pédagogiques telles que l'entraide parmi les apprenants du même groupe, les démonstrations en différé (*Replay Session*) ou en temps réel (*View Session*), l'assistance en mode asynchrone ou encore le support de l'évaluation des travaux des apprenants.

10.4.3. Création et exploitation de sessions partagées

Lorsqu'une session est partagée en mode lecture et écriture, deux apprenants ou plus peuvent y accéder pour effectuer les activités d'apprentissage d'une façon collaborative ; la Figure 10-15 illustre le partage d'une session ouverte par l'utilisateur *broisin* avec l'utilisateur *bouabid*. Cette fonction est la parfaite implémentation de l'*Awareness* de l'artéfact présenté dans les sections 2.3.8 et 4.7 car tous les acteurs prenant part à une session partagée agissent sur celle-ci et regardent en temps réel les actions des uns et des autres. Les sessions partagées constituent par ailleurs le meilleur mécanisme pour une assistance en temps réel assurée par le tuteur ou par des pairs.

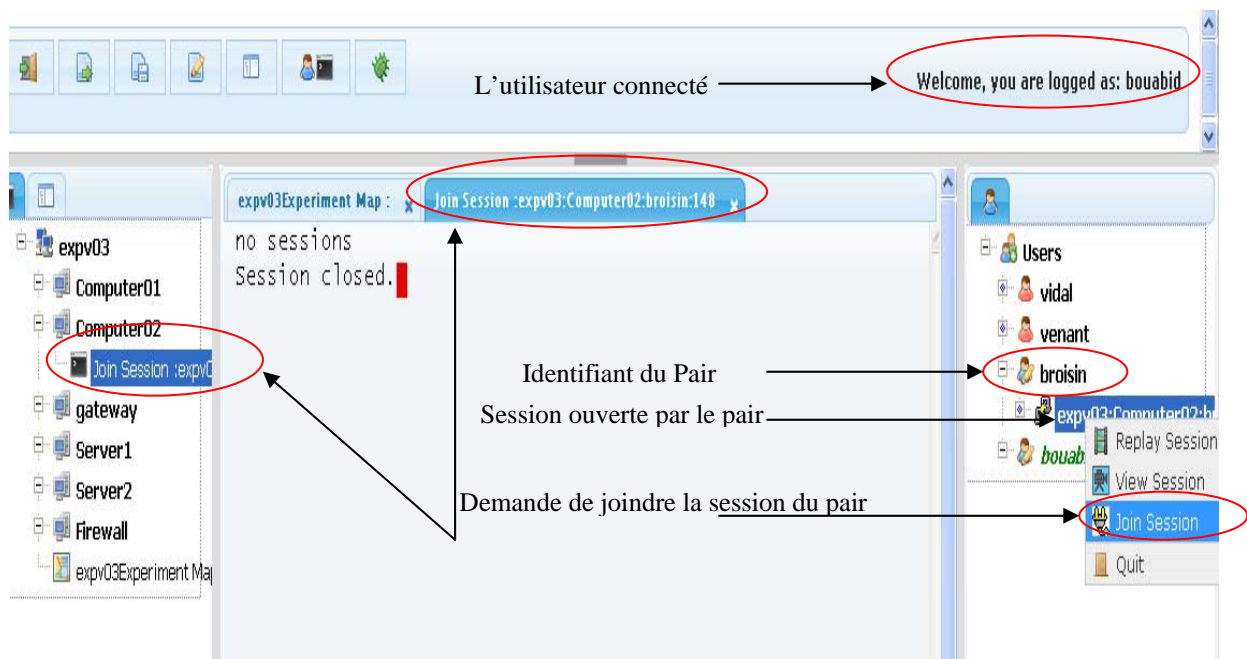


Figure 10-15. Accès partagé à une session ouverte en mode lecture et écriture.

10.4.4. L'outil de messagerie instantanée

L'objectif de l'outil de messagerie instantanée est de mettre à disposition des acteurs un moyen de communication synchrone « à portée de main ». En se situant aux côtés des outils d'exploration et d'exploitation d'une expérience, il encourage l'échange de connaissances ou d'idées, mais surtout facilite la coordination et l'organisation des activités afin d'atteindre les objectifs ciblés dans le cas d'une activité collaborative.

Fondé sur l'outil *open source AJAX Chat* à base de technologies web 2.0, cet outil a été configuré et intégré de sorte que chaque équipe d'apprenants et tuteurs dispose de quatre canaux (ou salons) de communication :

- Le salon Equipe, destiné à supporter le travail collaboratif entre les membres de l'équipe de manière informelle.
- Le salon Tutorat, dédié au suivi en ligne par le tuteur. Les apprenants y déposent leurs questions et sollicitations, puis accèdent aux réponses formulées par le tuteur.
- Le salon Public est accessible à tous les utilisateurs inscrits dans la plateforme, et peut être utilisé pour communiquer avec des acteurs ne faisant pas partie de l'équipe.

- Le salon « Nom-Utilisateur » est un salon privé à l'utilisateur lui-même, lui permettant d'écrire des notes ou des messages personnels.

La Figure 10-16 illustre en détail cet espace de communication où les acteurs sont encouragés à utiliser les boutons *d'émoticons* afin de symboliser les interventions les plus significatives, en particulier :

- « Main levée » pour attirer l'attention du tuteur afin de demander la parole pour poser une question ou solliciter une aide en cas de problème.
- « Point d'interrogation » pour attirer l'attention des pairs et du tuteur sur une question qui va être posée.
- « Point d'exclamation » pour indiquer qu'une intervention nécessite plus d'explications.
- « Idée » pour attirer l'attention des pairs et du tuteur sur une idée qui va être proposée.

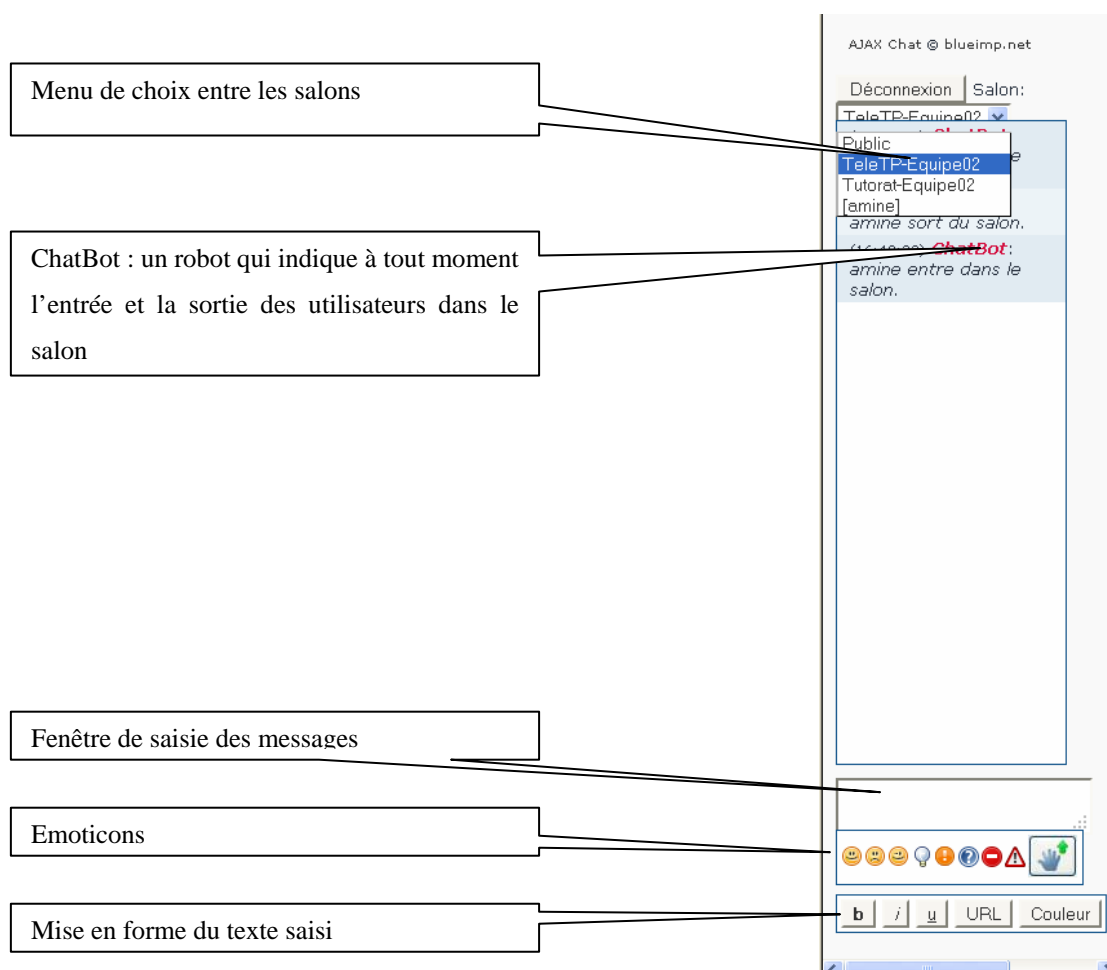


Figure 10-16. Détails de la fenêtre de messagerie instantanée.

Après avoir présenté l'espace de travail des apprenants, la prochaine section s'intéresse à l'espace de travail du tuteur qui offre des fonctionnalités supplémentaires supportant son activité de suivi en ligne.

10.5. Description de l'espace de travail du tuteur

L'espace de travail du tuteur offre tous les outils intégrés dans l'espace de travail des apprenants, complétés par d'autres fonctionnalités spécifiques à ses tâches et privilèges. Par exemple, un tuteur accède en mode lecture et écriture à toutes les sessions ouvertes par les apprenants, et dispose de tous les outils de tutorat offerts nativement par le LMS Moodle et qui sont liés aux activités traditionnelles de suivi de l'évolution des apprenants et d'évaluation de leurs travaux.

En outre, l'interface du tuteur au sein de notre IHM diffère de celle de l'apprenant par la présence d'un outil de supervision des expériences accessible via un onglet supplémentaire de la fenêtre de l'explorateur des expériences illustré par la Figure 10-17. Cet outil donne au tuteur des privilèges élargis sur l'expérience, en particulier grâce au menu contextuel associé à celle-ci et qui représente le plus haut niveau de l'arborescence. Certaines opérations du cycle de vie d'une expérience sont ainsi disponibles et présentées par la Figure 10-17 :

- Le démarrage (option *Start*) entraîne le démarrage de tous les nœuds composant l'expérience, y compris les commutateurs virtuels.
- L'arrêt (option *Stop*) provoque l'arrêt de tous les nœuds de l'expérience et des commutateurs virtuels.
- Le redémarrage (option *Reboot*) a pour conséquence le redémarrage de tous les nœuds de l'expérience.
- La réinitialisation (option *Reset*) entraîne l'arrêt des nœuds de l'expérience et leur redémarrage dans leur état d'origine.

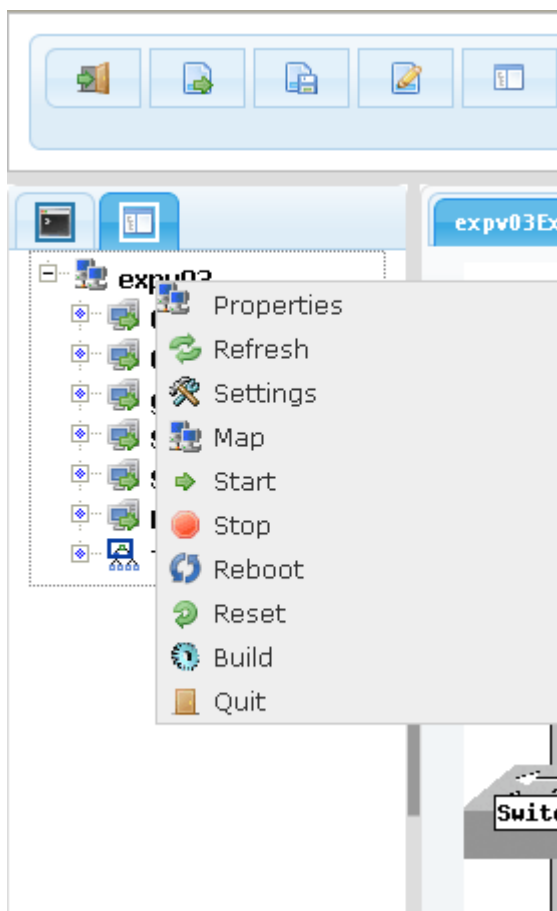


Figure 10-17. Menu contextuel associé à une expérience.

De la même manière que l'interface destinée aux apprenants, l'explorateur des expériences accessible aux tuteurs expose la liste des nœuds composant chacune des expériences. Une icône est associée à chacun des nœuds pour symboliser leur nature, et un indicateur de couleur indique leur état (vert lorsque le nœud est actif, rouge lorsque qu'il est inactif, et orange lorsque l'état est inconnu). Le tuteur dispose ainsi d'une vue globale de l'expérience et peut facilement identifier les apprenants en difficulté ou en situation de blocage lorsqu'un nœud ne se trouve pas dans un état « normal ».

En plus de proposer un menu contextuel pour effectuer les opérations présentées dans la section 10.3.2.1 (démarrage, arrêt, redémarrage, et réinitialisation des nœuds de l'expérience), le tuteur dispose d'une vue de chaque nœud d'un point de vue supervision : il peut visualiser l'ensemble des informations de gestion définies dans le modèle global de supervision des expériences en informatique présenté dans la section 9.3.2. Les éléments constituant un nœud

sont affichés de façon hiérarchique eux aussi, depuis les éléments globaux jusqu'aux éléments élémentaires. Sur la Figure 10-18, les composants d'une machine sont par exemple ses processeurs, le nom et la version de son système d'exploitation, le type et le point de montage de son système de fichier, ou la liste des interfaces réseau et leur configuration IP. Il est important de signaler ici qu'il ne s'agit que d'un exemple : c'est en fonction des objectifs de l'activité de télé-TP, des modèles CIM élaborés par les concepteurs et des Pilotes WBEM instrumentés que les informations ou indicateurs sont identifiés et présentés au tuteur.

En effet, ces informations sont obtenues en interrogeant le serveur WBEM via les services de la couche d'Intégration, et elles sont fournies par les Pilotes WBEM de l'outil MLN et ceux des machines virtuelles de l'expérience. Les informations sur la topologie réseau de l'expérience sont affichées à travers l'élément *TopologyGraph* qui représente l'instance CIM du même nom, et qui regroupe les différents commutateurs associés ; ces derniers correspondent aux instances *LANConnectivitySegment* du modèle global.

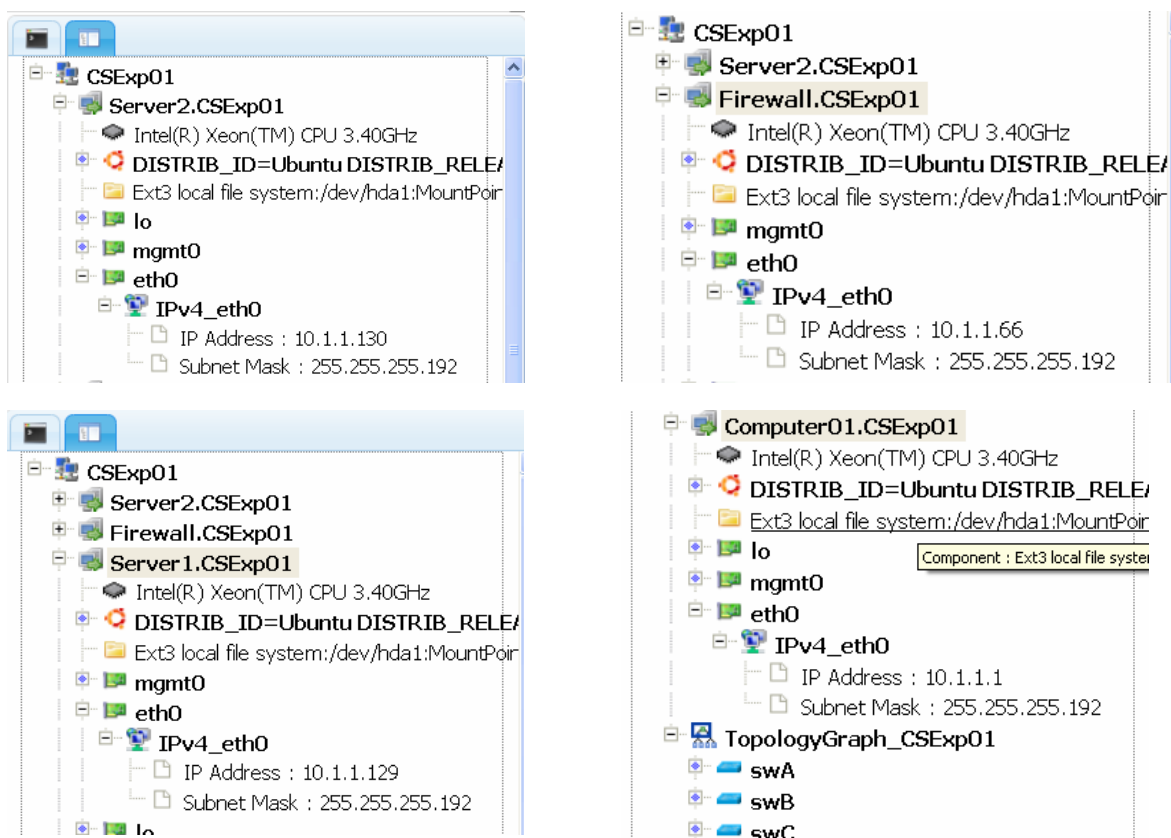


Figure 10-18. Supervision des nœuds d'une expérience.

10.6. Synthèse

Dans ce chapitre nous avons présenté les IHMs offertes par notre EIAH pour supporter les différentes activités pédagogiques des acteurs humains impliqués dans une activité de télé-TP. Ces interfaces exploitent les outils et services des couches d'Apprentissage et d'Intégration présentés dans le chapitre 9 pour rendre transparente la complexité des mécanismes mis en œuvre, et encourager l'intégration et le déploiement des activités de télé-TP au sein des cursus d'apprentissage en ligne.

Pour la conception et l'implémentation de ces IHMs fondées sur les technologies du web 2.0, nous avons essayé de satisfaire les objectifs et contraintes pédagogiques exprimés dans le chapitre 4 qui visent l'atteinte de l'efficacité pédagogique des télé-TPs. Ainsi, trois types d'interfaces ont été développés, correspondant aux trois principaux acteurs humains de notre EIAH : l'interface de conception technique et pédagogique, l'interface support à l'apprentissage, et l'interface dédiée au tutorat.

L'activité de conception est supportée par un ensemble d'IHMs permettant à l'enseignant de contrôler tout le cycle de vie des expériences intégrées dans ses cours en ligne. Nous avons implémenté une interface de conception visuelle des expériences réseau automatisant le processus de déploiement des expériences : le modèle d'instances CIM correspondant à l'objet pédagogique interactif « Expérience » est généré afin d'être injecté dans le serveur OpenPegasus de la couche d'Intégration, puis adapté à l'outil MLN afin d'assurer le déploiement des machines virtuelles et commutateurs nécessaires à l'expérience. L'objet pédagogique « Expérience » exprimé au format XML est accompagné d'une fiche de métadonnées au format LOM permettant son indexation dans un vivier d'objets pédagogiques comme celui de la fondation ARIADNE, et ainsi favoriser son partage et réutilisation au sein de la communauté.

L'interface d'apprentissage a été conçue en s'inspirant de la métaphore du « Cockpit » qui consiste à offrir une interface unique pour réaliser toutes les tâches. Afin de faciliter les activités d'apprentissage individuelles, nous nous sommes focalisés sur le développement d'interfaces d'exploration et de contrôle des expériences. Celles-ci s'appuient sur des outils utilisés dans les environnements de production pour offrir un sentiment de réalisme, et proposent des modes de contrôle visuels avancés afin de rendre la navigation ergonomique,

fluide et légère. L'espace de travail dédié aux activités d'apprentissage collaboratif est en grande partie supporté par le service de gestion des traces d'activités qui permet à tout moment de savoir qui est connecté, qui fait quoi et qui a fait quoi, mais il offre également un sentiment de présence sociale à travers la mise en œuvre des principes de l'*Awareness* présentés dans les sections 2.3.8 et 4.7.

Enfin, l'interface dédiée au tutorat fournit des privilèges plus élevés aux tuteurs qui peuvent consulter et intervenir sur les activités de n'importe quel apprenant, et implémente une interface de supervision offrant une vue globale et détaillée de l'ensemble des nœuds (et de leurs composants) des expériences. Le tuteur dispose ainsi de nombreuses informations sur l'évolution des apprenants, leurs activités et leurs productions. Ainsi nous avons mis en œuvre le principe socioconstructiviste, qui postule qu'un tuteur est un pair avancé collaborant avec un groupe d'apprenants.

Chapitre 11. Expérimentation

<u>11.1. Introduction</u>	270
<u>11.2. Démarche</u>	272
<u>11.3. Test d'utilisabilité</u>	273
<u>11.4. Résultats et interprétations</u>	276
<u>11.4.1. Observation du comportement des utilisateurs</u>	277
<u>11.4.2. Enquête de satisfaction</u>	278
<u>11.5. Synthèse</u>	279

11.1. Introduction

L'expérimentation de notre cadre de travail détaillé dans les deux chapitres précédents est une étape indispensable à la validation de nos propositions dans le domaine de l'enseignement de l'informatique d'une part, et de nos solutions génériques dont notre environnement représente une instantiation d'autre part. L'objectif de l'expérimentation consiste à vérifier que notre approche pour lever les verrous exprimés dans le chapitre 4 permet effectivement d'atteindre l'efficacité pédagogique des télé-TPs en s'assurant que l'apprentissage a eu lieu et que les objectifs pédagogiques attendus ont été atteints. L'expérimentation mettra également en avant les points forts de notre approche à consolider, ses points faibles à corriger, et apportera éventuellement d'autres pistes de recherche liées à de nouvelles problématiques.

Le public d'apprenants ayant pris part à notre expérimentation est constitué d'ingénieurs salariés en informatique dans le contexte de la formation continue. Deux équipes ont été recrutées : la première auprès du CERIST d'Alger⁸², la seconde auprès du service informatique et télé-enseignement de l'université Abou Bekr Belkaid à Tlemcen⁸³ (extrême Ouest de l'Algérie). Quatre apprenants constituent l'équipe du CERIST d'Alger, alors que la seconde équipe comprend six membres.

⁸² Centre de Recherche sur l'Information Scientifique et Technique, <http://www.cerist.dz>

⁸³ <http://www.univ-tlemcen.dz>

L'activité d'apprentissage dans sa globalité correspond au scénario type présenté dans le chapitre 9, elle est accessible sous la forme d'un cours Moodle respectant la structure illustrée par la Figure 9-24. Chaque équipe dispose de son propre environnement de télé-TP, et en particulier de sa propre expérience. En effet, grâce à la souplesse de notre banc d'essais, une nouvelle instance de l'expérience type de notre télé-TP est créée pour chaque équipe.

Le serveur physique hébergeant la plateforme d'expérimentation (fondée sur la technologie de virtualisation XEN et l'outil MLN) est une machine à deux processeurs INTEL Xeon dotée de 2GO de mémoire vive et quelques 200GO d'espace de stockage ; le système d'exploitation installé sur cette machine est Linux Ubuntu Server. Le serveur n'étant pas dédié uniquement à notre expérimentation, nous n'avons pu déployer que trois instances de notre expérience type, ce qui représente au total 18 machines virtuelles. Notons que les serveurs « nouvelle génération » peuvent supporter un nombre de machines virtuelles beaucoup plus important tout en assurant des performances très proches de celles du matériel, puisqu'ils peuvent être munis de nombreux processeurs de plus en plus denses (jusqu'à 8 cœurs par processeur) et embarquer une quantité de mémoire vive largement supérieure à celle installée dans notre serveur d'expérimentation. Toutefois, une architecture optimale consisterait à concevoir l'environnement du laboratoire avec une solution de stockage de masse (de type baie de stockage par exemple) qui soit partagée entre plusieurs serveurs physiques pour offrir l'espace disque nécessaire aux expériences les plus gourmandes en machines virtuelles, mais également afin d'implémenter des stratégies de sauvegarde et de migration en temps réel de ces machines.

Etant données les contraintes de disponibilité de matériel, dans le cadre de notre expérimentation, la plateforme Moodle ainsi que le serveur OpenPegasus ont été installés sur la même machine physique que la plateforme d'expérimentation. Cependant, les échanges entre les différentes couches de notre système global fondés sur des protocoles de communication réseau offrent l'opportunité de distribuer les trois sous-systèmes sur des machines physiques géographiquement délocalisées.

Cette première expérimentation a été organisée en deux périodes consécutives de 15 jours (du 1^{er} au 15 juillet 2011, puis du 1^{er} au 15 Août de la même année), dédiées chacune à chaque équipe, suivant les contraintes des expérimentateurs. Les ingénieurs des deux équipes

n'étaient pas complètement distants les uns des autres puisqu'ils travaillent tous dans la même institution, mais chacun a participé à la réalisation du télé-TP à partir de son poste de travail dans son bureau, cloisonné des autres participants.

Pour le suivi de ces deux sessions, nous étions obligé d'assurer nous même le tutorat vu l'indisponibilité de volontaires en cette période de l'année ; nous avons assuré cette tâche complètement à distance par rapport à la deuxième équipe.

Les consignes formulées à l'introduction de l'énoncé de l'activité de télé-TP (voir les détails dans l'annexe F) indiquent qu'elle doit être réalisée d'une manière collaborative, mais n'imposent pas de méthode précise pour cette collaboration : les apprenants-testeurs doivent s'entendre sur le meilleur moyen de collaborer en fonction des contraintes temporelles de chacun. Toutefois, nous avons suggéré la démarche suivante : travailler d'une manière collaborative sur les questions posées dans l'énoncé de l'activité, puis passer à la prise en main de l'expérience en se partageant les machines virtuelles.

11.2. Démarche

Notre démarche est fondée sur l'analyse de données empiriques issues de deux types de sources : une enquête menée auprès des acteurs de la formation (essentiellement des apprenants), et l'observation du comportement de ces acteurs au cours des sessions d'expérimentation à travers l'analyse des différents types de traces d'activités enregistrées.

L'enquête repose sur la technique d'évaluation heuristique appelée test d'utilisabilité (*Usability Testing* ; Nielson, 1993), dont l'objectif consiste à mesurer les degrés d'utilité, d'utilisabilité et d'acceptabilité d'une application informatique par les utilisateurs finaux. A partir des heuristiques de Nielson, nous avons élaboré un questionnaire d'utilisabilité présenté dans la section suivante que doit remplir chaque apprenant-testeur et dans des expériences futures par les enseignants-testeurs. Nous n'avons pas construit deux questionnaires différents par catégorie d'acteur pour la simple raison que la majorité des fonctionnalités offertes par notre application est disponible aux deux types d'acteurs, à l'exception de l'outil de supervision global de l'expérience.

L'observation du comportement des acteurs de l'expérimentation repose quant à elle sur les capacités de traçage offertes nativement par notre plateforme. Nous nous sommes focalisés

dans un premier temps sur une analyse quantitative des traces recueillies (nombre de sessions ouvertes, nombre de commandes exécutées, statistiques d'utilisation de chaque outil, etc.). Nous avons exclu l'analyse comportementale qualitative (contenu des commandes, dépendances entre les actions, etc.), vus les moyens et le temps importants qu'elle pourrait consommer. Un « survol » des traces d'activités permettra tout de même d'identifier des contradictions sur les réponses obtenues lors de l'enquête.

11.3. Test d'utilisabilité

Le questionnaire d'utilisabilité soumis aux acteurs de l'expérimentation comporte un ensemble de dix principes d'évaluation générés à partir de quatre critères : (1) la facilité de l'apprentissage de l'IHM, (2) l'efficacité de l'exploitation de l'IHM, (3) la gestion des erreurs, et (4) la satisfaction des utilisateurs. Les principes tirés par Nielson sont (Nielson, 2005 ; traduit par Bétrancourt et Benetos, 2010) : «

1. **Visibilité du statut du système** : le système devrait maintenir l'utilisateur toujours informé sur sa localisation. Les utilisateurs doivent savoir où ils se trouvent et où est-ce qu'ils doivent aller.
2. **Correspondance entre le système et le monde réel** : le système devrait "parler" la langue des utilisateurs de telle façon que les informations apparaissent dans un ordre naturel et logique, en se conformant aux mots, expressions et conventions usuels.
3. **Contrôle de l'utilisateur et liberté** : le système devrait être très "obéissant" aux actions de l'utilisateur et offrir des "sorties de secours" clairement marquées pour les opérations exécutées mais non désirées, ainsi que des options pour faire/défaire les opérations. Il faut aussi permettre à l'utilisateur de personnaliser sa zone de travail comme la couleur, la police de caractères, la largeur de l'écran, la version du navigateur, etc.
4. **Consistance et étalon (normalisation)** : suivre les conventions de plateforme est la façon la plus sûre de respecter ce principe. Les utilisateurs ne doivent pas se demander si les différents mots, situations ou actions veulent dire la même chose. Les titres et entêtes doivent être représentatifs du contenu.
5. **Prévention contre les erreurs** : le système devrait prévoir les erreurs que l'utilisateur peut commettre. Il doit offrir des messages d'erreurs pertinents et des instructions compréhensibles pour les corriger afin d'éviter que des problèmes surviennent.

6. **Reconnaissance plutôt que rappel** : le système devrait rendre les objets, les actions et les options visibles. Les directives pour l'usage du système doivent être visibles ou facilement récupérables à chaque fois que c'est nécessaire.
7. **Flexibilité et efficacité d'utilisation** : le système devrait offrir des raccourcis pour les utilisateurs expérimentés afin d'accélérer l'exécution des actions les plus fréquemment utilisées.
8. **"Design" esthétique et minimaliste** : les dialogues devraient contenir seulement les informations pertinentes et principales pour les rendre bien visibles. Chaque unité d'information supplémentaire dans un dialogue rivalise avec les unités pertinentes et diminue leur visibilité relative.
9. **Fonctions de l'aide (reconnaissance, diagnostic et récupération d'erreurs)** : les messages d'erreurs devraient être exprimés dans la langue ordinaire, sans code en indiquant précisément le problème de manière constructive, claire et compressible et de manière à offrir une solution aux problèmes, ou en tout cas à aider à les résoudre.
10. **Aide et documentation.** »

Ces heuristiques constituent les rubriques de base de notre enquête en ligne, en plus de deux rubriques supplémentaires destinées à évaluer la qualité de l'apprentissage et le tutorat. La majorité des questions proposées dans ces rubriques sont de type choix multiples, avec une réponse unique sur une échelle à cinq niveaux : « Neutre, Pas d'accord, Plutôt d'accord, D'accord, Tout à fait d'accord ». Ce genre de question facilite la tâche aux utilisateurs, mais favorise également l'analyse a posteriori par la génération de statistiques et la création de graphiques par exemple. Seule la douzième et dernière question est de type réponse libre, afin de recueillir l'avis de l'apprenant/tuteur par rapport à des aspects qui n'auraient pas été pris en charge par les questions précédentes. Le Tableau 11-1 résume le questionnaire proposé.

Tableau 11-1. Résumé du questionnaire d'enquête sur l'utilisabilité.

Rubrique	Question
1. Visibilité de l'état de l'expérience	<ul style="list-style-type: none"> • L'explorateur de l'expérience permet de repérer rapidement les machines de l'expérience et les sessions qui y sont ouvertes. • L'explorateur des utilisateurs permet de repérer rapidement les autres membres de l'équipe et l'enseignant, ainsi que les sessions qu'ils ont ouvertes.

	<ul style="list-style-type: none"> • L'historique des sessions permet de s'informer sur les actions effectuées par l'utilisateur lui-même ou par les autres, afin de comprendre l'état actuel de l'expérience.
2. Réalisme de l'expérience	<ul style="list-style-type: none"> • L'expérience en ligne proposée correspond en tous points aux expériences déployées dans les laboratoires classiques. • L'interface Terminal "Ligne de commande" (Console ou SSH) est réaliste et correspond aux interfaces utilisées sur des systèmes réels.
3. Contrôles et libertés d'utilisation	<ul style="list-style-type: none"> • Le système permet de revenir dans un état cohérent de l'expérience en cas d'erreur. • En particulier, l'accès en mode Console sur une machine permet de remettre cette dernière dans un état cohérent après une fausse manipulation. • L'IHM est souple en terme de personnalisation de l'affichage (dimensionnement des fenêtres, cacher/montrer les fenêtres, couleurs, etc.).
4. Correspondance des symboles de l'interface avec ceux des outils existants	<ul style="list-style-type: none"> • Les symboles graphiques et les contrôles de l'IHM (comme l'exploration en arborescence, les menus contextuels et les onglets) correspondent aux symboles et contrôles courants dans les autres systèmes/logiciels.
5. Erreurs et <i>bugs</i>	<ul style="list-style-type: none"> • L'explorateur de l'historique des sessions de l'utilisateur connecté (et des autres utilisateurs, qu'ils soient apprenants ou enseignants) permet d'identifier les fausses manipulations et de les corriger.
6. Rapidité d'apprentissage	<ul style="list-style-type: none"> • D'une manière générale, l'interface d'expérimentation est facile à prendre en main. • L'interface d'expérimentation rend visible les composants de l'expérience, les utilisateurs et les sessions, et en facilite le repérage.
7. Flexibilité et efficience d'utilisation	<ul style="list-style-type: none"> • Les menus contextuels permettent de créer rapidement des sessions de travail sur les nœuds de l'expérience et d'y accéder. • Les onglets, sur la fenêtre centrale de travail, associés aux éléments de l'explorateur de l'expérience, permettent de naviguer rapidement vers la session qui correspond à un dispositif particulier.
8. Qualité du <i>design</i>	<ul style="list-style-type: none"> • L'interface affiche les fonctions essentielles à la gestion d'expériences, et n'est pas encombrée de barres de menus ou d'options perturbant la réalisation des activités primordiales à la conduite d'un télé-TP.
9. Fonctions de l'aide (reconnaissance, diagnostic et récupération d'erreurs)	<ul style="list-style-type: none"> • La fonction de partage des sessions en temps réel avec un ou plusieurs co-équipiers et/ou le tuteur favorise l'aide en temps réel. • La fonction de consultation de l'historique des sessions permet

	d'apprendre à partir des actions des autres.
10. Aide et documentation	<ul style="list-style-type: none"> • L'outil de communication fourni (la messagerie instantanée) est suffisant pour obtenir de l'aide en ligne.
11. Activité d'apprentissage/tutorat	<ul style="list-style-type: none"> • D'une manière générale, je suis satisfait de ce système. • L'outil de messagerie instantanée, l'explorateur des utilisateurs ainsi que l'explorateur des sessions Terminal permettent de savoir en temps réel quel utilisateur est connecté et les opérations qu'il effectue. • La messagerie instantanée facilite la coordination efficace des actions des membres de l'équipe. • La messagerie instantanée permet de demander de l'aide aux co-équipiers ou au tuteur en ligne, et de l'obtenir efficacement. • La consultation en temps réel ou en différé de la session d'un autre co-équipier ou du tuteur permet d'apprendre de nouvelles connaissances/compétences ou de surmonter des situations de blocage. • Les sessions partagées permettent de rendre efficace le travail d'équipe et de réaliser rapidement les buts de l'activité.
12. Appréciation générale	<ul style="list-style-type: none"> • Donner, en toute objectivité, votre avis général sur le système et son IHM en indiquant leurs points forts, leurs points faibles, ainsi que vos avis/conseils pour améliorer l'interface.

11.4. Résultats et interprétations

La plupart des recherches sur le thème des télé-TPs insistent sur la nécessité de mener des expériences à échelle réelle sur le plus grand nombre d'apprenants possible, afin de valider les solutions proposées et d'identifier les points faibles à améliorer (Ma et Nickerson, 2006 ; Nickerson et al., 2007 ; Corter et al., 2007 ; Zacharia et Olympiou, 2010). Dans notre projet ceci n'a pu être réalisé à ce jour, principalement en raison des contraintes de moyens et de temps, mais aussi de la difficulté à recruter des apprenants. Néanmoins, la finalisation et le déploiement de nos propositions au sein de l'IUT A Paul Sabatier de l'Université de Toulouse sont en cours, et des expérimentations avec des étudiants de 1^{ère} année en informatique verront le jour au 2^{ème} semestre de l'année universitaire à venir. Aussi, l'expérimentation que nous avons déjà menée donne un aperçu de la pertinence de nos propositions et met en lumière de nouvelles problématiques.

11.4.1. Observation du comportement des utilisateurs

Deux types de traces ont pu être recueillis lors de notre expérimentation : (1) les fichiers journaux générés par le LMS Moodle qui constitue le point d'entrée à toute l'activité d'apprentissage, et en particulier à l'expérience en ligne, et (2) les traces liées à l'application de télé-TP que nous avons développée et intégrée à Moodle.

En ce qui concerne la plateforme Moodle, nous avons constaté des interactions importantes avec les différentes ressources mises à disposition pour l'activité du Télé-TP. A titre indicatif, pour l'ensemble des deux équipes, les statistiques globales suivantes ont été constatées : 120 accès à la plateforme, 258 consultations au cours renfermant les différentes ressources, 83 accès aux ressources (support de cours, guides techniques, etc.), et 92 accès à l'IHM de l'expérience. Notons que très peu d'accès aux outils de communication natifs à Moodle ont été relevés : aucun accès à l'outil de messagerie instantanée, et seulement deux accès à l'outil Forum. Ces statistiques peuvent s'expliquer par le fait que les apprenants, qui sont des ingénieurs salariés, n'ont pas eu besoin de l'outil de messagerie instantanée pour s'entraider à comprendre le contenu des cours et des ressources, mais également par le fait que la période de test était trop courte pour une utilisation efficace de l'outil Forum ; enfin, notons que les testeurs n'ont pas reconcé à leurs outils de messagerie préférés au profit de celui offert par Moodle. Ceci nous permet d'affirmer que les expérimentateurs n'ont pas rencontré de difficulté particulière à utiliser ce LMS, les fichiers journaux renforçant ce sentiment puisque l'activité de télé-TP dans sa globalité a été parcourue dans l'ordre voulu (Pré-Lab, Lab, Post-Lab).

Au niveau de l'application d'expérimentation que nous avons proposée, une très dense activité a été constatée : les apprenants n'ont pas éprouvé de difficulté à manipuler et à adopter l'outil. Par exemple, pour la première équipe, 29 sessions ont été ouvertes en moyenne par utilisateur, à raison d'une centaine de commandes par session (2545 commandes au total), alors que nous avons constaté une moyenne de 14 sessions ouvertes par apprenant de la deuxième équipe (774 commandes au total). De nombreuses interactions ont également pris place avec l'outil de messagerie instantanée intégré à notre application : les membres de la première équipe ont utilisé l'outil 51 fois, et 400 interventions réparties sur les trois salons (Public, Equipe et Tutorat) ont été relevées contre 23 connexions à l'outil et 120 interventions

pour la deuxième équipe. Notons également qu'une séance de tutorat en ligne par équipe a été tenue afin d'apporter de l'aide à des apprenants en difficulté.

D'autre part nous avons pu vérifier que le système de gestion des traces d'activités, initialement conçu à des fins pédagogiques, s'est révélé un puissant moyen pour évaluer toute la plateforme. En effet, les projets similaires ont souvent recours aux outils de « *Screencast* » pour enregistrer et dérouler en temps réel ou a posteriori le comportement d'utilisateurs faisant face à une application. Outre les contraintes d'installation de ces outils sur les postes de travail des expérimentateurs, les utilisateurs peuvent délibérément arrêter le fonctionnement de ces systèmes à tout moment. Dans notre approche, aucun logiciel additionnel n'est requis sur le poste des apprenants et tuteurs puisque les outils et mécanismes nécessaires au traçage de leurs activités sont intégrés dans l'application d'expérimentation.

11.4.2. Enquête de satisfaction

Les premiers résultats de l'enquête obtenus auprès des utilisateurs sont encourageants et confortent notre interprétation de l'analyse des traces présentée dans la section précédente. En effet, près de la moitié des réponses (49,68%) exprime la satisfaction des sujets, l'autre moitié comprenant 22,98% de satisfaction totale et 16,77% de satisfaction moyenne. Seulement 12% des réponses expriment une position neutre ou non satisfaisante. L'Annexe G donne les détails des résultats de chaque question sous forme de pourcentages et de graphiques.

Les utilisateurs sont particulièrement satisfaits de la visibilité de l'expérience et des moyens de son exploration, grâce notamment aux informations rendues disponibles par l'explorateur de l'expérience et celui des utilisateurs. Ils sont également très satisfaits de la fenêtre Terminal qui assure le réalisme des moyens de télé-instrumentation.

Au niveau du *design*, ils considèrent que l'interface est assez souple en terme de personnalisation de l'affichage, et que les symboles graphiques familiers simplifient l'identification des fonctionnalités associées.

Ils considèrent également que la consultation de l'historique des sessions est un bon moyen pour identifier les manipulations à l'origine d'erreurs ou de situations incohérentes. Cette fonctionnalité est également considérée comme un bon moyen pour apprendre des autres, au

travers de la consultation de sessions « idéales » ou de consultation en temps réel de sessions partagées entre plusieurs acteurs.

Les sujets ont aussi exprimé leur satisfaction quant à la prise en main rapide de l'application, ainsi que l'efficacité des outils offerts dans la réalisation des tâches attendues. Toutefois, les réponses relatives aux fonctions de contrôle avancé des expériences (retour en arrière, réinitialisation, etc.) reflètent le fait que les utilisateurs n'ont pas eu (ou n'ont pas pris) assez de temps pour les expérimenter.

Les réactions par rapport au *design* de l'application ne sont pas du même niveau, mais il représente un bon compromis entre la complexité des fonctions et l'encombrement de l'interface par les éléments associés à ces fonctions.

En ce qui concerne les activités d'apprentissage et de tutorat, la plupart des acteurs est totalement satisfaite du système et considère que le travail d'équipe et le soutien en ligne sont bien supportés par l'implémentation des principes de l'*Awareness*, en particulier l'*Awareness* de l'*Artefact*. Ils considèrent aussi que l'outil de communication synchrone est un bon moyen de communication et de collaboration en temps réel au cours d'une expérimentation.

En plus des résultats ci-dessus, les réponses libres à la dernière question nous réconfortent mais attirent notre attention sur certains problèmes à solutionner, en particulier :

- Ajouter d'autres fonctions collaboratives telles que la demande de prise de contrôle d'une session même si elle est partagée.
- Ajouter des fonctions d'auto-évaluation comme une barre de progression qui indique l'état d'avancement de l'apprenant ou du groupe d'apprenants dans la réalisation de l'activité.
- Permettre la consultation accélérée des sessions afin de gagner du temps, à la manière d'un lecteur multimédia ; l'interface actuelle ne permet que la consultation selon le rythme de l'utilisateur qui a réalisé la session.

11.5. Synthèse

Ce chapitre a présenté l'expérimentation de notre cadre de travail dans des conditions réelles. L'évaluation est fondée sur l'analyse quantitative de données empiriques issues de deux sources : les résultats d'une enquête d'utilisabilité de notre plateforme auprès des utilisateurs

(apprenants et tuteurs), et l'analyse quantitative de leur comportement face aux outils de la plateforme.

Les résultats de l'enquête d'utilisabilité qui repose essentiellement sur l'heuristique de Nielson sont satisfaisants et nous encouragent à poursuivre cette recherche. En effet les apprenants ont apprécié les outils mis à leur disposition par l'application, et en particulier les fonctions d'identification et de repérage des composants de l'expérience, des utilisateurs et de leur(s) session(s) de travail. Ils ont cependant rencontré certaines difficultés liées essentiellement à l'instabilité du système développé. En effet, faute de temps, certaines fonctionnalités prévues n'ont pas été implémentées (gestion de tous les cas d'erreurs, régulation de l'accès aux sessions partagées, etc.), sans parler de la documentation et des manuels d'aide couvrant tous les aspects et toutes les situations.

La consultation des traces d'activités a révélé une activité intense renforçant les résultats de l'enquête. Nous avons constaté avec satisfaction l'utilité de la majorité des fonctions offertes, à travers les fichiers journaux nativement intégrés à la plateforme Moodle d'une part, et les traces d'activités issues de l'IHM d'expérimentation d'autre part. Le scénario du télé-TP a été déroulé dans l'ordre voulu : les ressources mises en ligne via la plateforme Moodle ont été consultées malgré l'engouement exprimé vis-à-vis de l'interface d'expérimentation.

En tant que tuteurs, nous avons pu confirmer la pertinence des outils disponibles, à commencer par les fonctions de consultation des sessions en cours, terminées ou partagées, mais également des fonctions de supervision globale de l'expérience qui exposent l'évolution de ses propriétés. L'analyse de l'utilisation des outils de collaboration et de communication montre une exploitation minime des outils de messagerie instantanée et de forum de discussion offerts nativement par la plateforme Moodle. En revanche, l'outil de messagerie instantanée intégré à l'IHM d'expérimentation a été pleinement exploité, ce qui confirme l'hypothèse de la nécessité de l'intégration de tels outils au sein des interfaces d'expérimentation pour une collaboration accrue et un suivi efficace des acteurs.

Dans le futur, une analyse comportementale plus poussée des utilisateurs doit être conduite afin d'obtenir des résultats et interprétations plus précis. Il s'agit effectivement de l'enjeu futur de cette recherche : exploiter les traces d'activités enregistrées pour analyser, de manière automatisée, les comportements des utilisateurs afin d'identifier les lacunes et y apporter les

aides adéquates telles qu'un système de tutorat intelligent. Le déploiement de notre approche dans d'autres contextes d'études et disciplines contribuera également à la validation de nos propositions génériques.

Conclusion

Chapitre 12. Conclusions et perspectives

<u>12.1. Bilan général</u>	284
<u>12.1.1. Introduction et normalisation de l'objet pédagogique « Expérience »</u>	285
<u>12.1.2. Originalités et avantages des expériences modélisées en CIM</u>	287
<u>12.1.3. La couche d'Intégration : un moteur générique pour l'opérationnalisation des expériences</u>	288
<u>12.1.4. Des modèles génériques pour la gestion des expériences et des traces.</u>	289
<u>12.1.5. Outils et IHMs supports à l'apprentissage et au tutorat</u>	290
<u>12.2. Perspectives</u>	291
<u>12.2.1. Perspectives générales</u>	291
<u>12.2.2. Perspectives spécifiques au domaine informatique</u>	295

12.1. Bilan général

Ce travail de thèse qui s'inscrit dans le domaine des EIAHs est une contribution à la résolution de la problématique générale de mise en ligne de travaux pratiques avec une application au contexte particulier de l'enseignement de la discipline informatique. Nous avons tenté au cours de ce travail de proposer une architecture générique qui intègre les laboratoires d'expérimentation aux EIAHs existants, indépendamment de la discipline enseignée. Nous avons proposé un ensemble de modèles et de méthodes qui encadrent les activités d'apprentissage et d'enseignement de télé-TPs en réponse à cette problématique. Enfin nous avons instancié cette architecture et ces modèles pour réaliser un EIAH spécifique aux télé-TPs en informatique.

Pour atteindre ces résultats, nous nous sommes focalisés sur deux problématiques sous-jacentes :

- Comment mettre en ligne des activités de télé-TP tout en atteignant l'efficacité pédagogique des TP traditionnels ?
- Comment intégrer les télé-TPs dans le processus général de normalisation des objets pédagogiques et des plateformes de télé-enseignement ? Les enjeux de partage et

d'échange des expériences, de mutualisation des ressources partagées, de minimisation des coûts par la facilité d'accès sont ici abordés.

A partir de la revue de littérature traitant du même sujet d'étude, nous avons identifié un certain nombre de contraintes à lever pour atteindre cette efficacité pédagogique :

- Des IHMs d'expérimentation réalistes comparées aux outils de commandes d'un laboratoire classique, qui rendent compte fidèlement des comportements, de l'évolution et des événements survenus sur le laboratoire distant.
- L'intégration des activités de télé-TP dans une démarche de conception pédagogique qui prend en compte les théories modernes d'apprentissage (en particulier le socioconstructivisme et l'apprentissage situé), et la définition dès le départ d'objectifs pédagogiques à atteindre clairement déterminés.
- Construire un dispositif d'apprentissage contenant des outils et IHMs dédiés à la communication, à l'expérimentation collaborative en ligne et au tutorat, aussi bien en mode synchrone qu'asynchrone.

12.1.1. Introduction et normalisation de l'objet pédagogique « Expérience »

Les travaux actuels sur la normalisation des télé-TPs proposent une double structuration : le scénario d'activité d'apprentissage (généralement exprimé avec un langage de modélisation pédagogique normalisé comme IMS-LD) et le laboratoire support à l'activité de télé-TP (modélisé par un langage standard et ouvert comme les ontologies). Le modèle descriptif du laboratoire est souvent constitué d'une partie opérative décrivant sa logique de fonctionnement et d'une partie commande qui reflète son interface de télé-opération. Le modèle du laboratoire est ensuite opérationnalisé par un composant logiciel qui d'une part assure la connexion avec les moyens de commande du laboratoire réel, et d'autre part expose aux utilisateurs l'IHM de télé-opération. En définitif, l'objet pédagogique « Laboratoire en ligne » est un monolithe comprenant son modèle descriptif et son implémentation.

Toutefois, diverses considérations pédagogiques et techniques doivent être prises en compte :

- Les définitions relatives aux activités de travaux pratiques décomposent ce genre d'activité en trois composants imbriqués : l'*activité d'apprentissage* qui utilise un *laboratoire* pour mener une *expérience*.
- Un même laboratoire peut servir à plusieurs expériences, et donc plusieurs TPs, car l'expérience est définie comme l'ensemble des conditions et paramètres initiaux d'un laboratoire.
- Avec le développement des technologies de grilles de calcul et de co-laboratoires, une expérience peut être distribuée sur plusieurs laboratoires distants lorsque les ressources nécessaires ne sont pas réunies dans le même laboratoire.

Par conséquent notre proposition est fondée sur une structuration plus fine des télé-TPs, et une opérationnalisation qui sépare le contenu (modèle descriptif pédagogique et technique) du contenant (IHMs, composants et outils logiciels d'exploitation, etc.).

Nous avons introduit un nouveau type d'*objet pédagogique interactif*, « l'*expérience en ligne* », sous la forme d'un objet pédagogique LOM standard caractérisé par une description pédagogique spécifique permettant son insertion et exploitation par les EIAHs, qui intègre une description technique fondée sur le méta-modèle normalisé CIM dont le rôle est d'en permettre le déploiement et la gestion au niveau des laboratoires distants. Ainsi, selon notre approche, l'objet « expérience en ligne » ne contient que le modèle descriptif technique (modèle CIM en format XML) imbriqué dans le modèle descriptif pédagogique (fiche LOM en format XML), afin d'indiquer les propriétés et opérations nécessaires à son déploiement et à son exploitation par des outils de contrôle et de supervision tiers.

Le choix du méta-modèle CIM pour la description technique des expériences en ligne se justifie non seulement par le lien étroit avec le domaine de la gestion de réseaux et de systèmes au niveau du contrôle et de la supervision de dispositifs distants, mais également par l'existence de l'architecture informatique normalisée WBEM qui opérationnalise ce méta-modèle. L'architecture WBEM, qui constitue le cœur de la couche d'Intégration de l'architecture global de notre EIAH, assure une exploitation unifiée et homogénéisée des objets CIM, et donc des entités réelles qu'ils modélisent.

Plus précisément, notre proposition consiste à modéliser trois niveaux d'objets impliqués dans la construction d'expériences en ligne, en plus d'un modèle spécifique dédié au suivi des activités des différents acteurs humains impliqués dans une activité de télé-TP :

- Le laboratoire (en ligne) en tant qu'entité administrative appartenant à une institution et offrant un certain nombre de ressources et de services d'exploitation de ces ressources (en particulier la gestion de la réservation et de l'allocation de ces ressources).
- Les ressources mises à disposition par chaque laboratoire (équipements, outils de mesure, outils de commandes, outils d'observation, logiciels spécifiques, etc.) qui sont les briques élémentaires utilisées pour la conception, le déploiement et l'exploitation des expériences.
- Les expériences (en ligne) en tant qu'entités plus ou moins complexes, composées d'un certain nombre de ressources offertes par un ou plusieurs laboratoires. Elles sont construites en deux phases : (1) la sélection des ressources que l'enseignant concepteur juge nécessaires à la construction de son expérience parmi celles offertes par le(s) laboratoire(s) distant(s) (par exemple un ensemble de machines virtuelles), puis (2) la modélisation proprement dite de l'expérience qui consiste à initialiser ses composants et éventuellement établir des connexions entre elles (comme la construction de liens réseaux entre les machines sélectionnées).
- Le modèle des traces d'activités, dont l'objectif est d'enregistrer finement les opérations et actions effectuées par les acteurs du télé-TP dans des bases de données externes, dédiées aux traces d'activités enregistrées sur tout type d'objet pédagogique. Le suivi efficace des activités humaines dans les EIAHs pour analyser les usages et les comportements des différents acteurs distants est nécessaire à la réingénierie des EIAHs et des scénarii d'apprentissage.

Le résultat de cette modélisation est un objet pédagogique qui peut être stocké et indexé dans les viviers d'objets pédagogiques assurant sa disponibilité, sa pérennité et son partage par la communauté des acteurs de la formation et l'enseignement.

12.1.2. Originalités et avantages des expériences modélisées en CIM

Une originalité de notre contribution réside dans l'indépendance de l'objet « Expérience » vis-à-vis de son implémentation et de l'IHM utilisateur qui l'exploite. Pour opérationnaliser

l'expérience, nous avons construit une architecture trois-tiers qui assure une intégration transparente entre EIAHs et laboratoires distants via une couche intermédiaire de traitements fondée sur l'architecture WBEM. Quant à l'IHM de contrôle et de supervision des expériences, elle peut être déléguée à une autre équipe de développement spécialisée qui exploite les services web exposés par cette couche intermédiaire. Comme exemple, nous avons développé une IHM de contrôle des expériences en informatique reposant sur les terminaux textuels, mais nous aurions pu opter pour un autre type d'IHM (interfaces graphiques à partir des technologies de réalité virtuelle, de réalité mixte, etc.). Les avantages de cette approche sont multiples :

- Le concepteur pédagogique se focalise uniquement sur la modélisation de ses expériences et peut en déléguer l'opérationnalisation et l'IHM à une autre équipe de développement. Mieux encore, dans notre contexte d'étude, nous avons développé une IHM unique pour toutes les expériences de notre domaine et l'opérationnalisation du modèle générique des expériences permet de prendre en charge automatiquement tous les modèles dérivés.
- Le modèle en format textuel est aisément portable d'un système à un autre, et surtout facilite la mise en marche de l'objet pédagogique « Expérience » dans le processus de normalisation à travers son indexation dans un vivier de connaissances par exemple.
- L'expérience présente un cycle de vie itératif comprenant les phases de modélisation, construction, (re)démarrage, arrêt, sauvegarde et suppression ; notre cycle de vie des expériences est complété par une phase de description LOM en étroite relation avec le cycle du télé-TP associé.
- Le modèle étant indépendant d'un laboratoire particulier, il peut s'adapter à différentes technologies sous-jacentes. Par exemple, dans notre implémentation, le modèle d'une expérience informatique peut être associé à des machines physiques, virtuelles, ou à une installation mixte.
- Enfin, l'expérience peut être déployée sur plusieurs laboratoires co-localisés ou distribués.

12.1.3. La couche d'Intégration : un moteur générique pour l'opérationnalisation des expériences

La généricité de la couche d'Intégration rend notre proposition valable à toutes les disciplines, indépendamment de l'EIAH et du laboratoire cibles. Chaque EIAH est spécifique au niveau

de son IHM d'expérimentation et du laboratoire qu'il exploite. Nous avons rappelé plus haut que les services web proposés par notre couche d'Intégration offrent une variété de possibilités quant au développement d'IHMs dédiées aux télé-TPs. D'autre part, la mise en œuvre de la communication entre le laboratoire et cette couche intermédiaire consiste à déclarer les modèles spécifiques au domaine visé dans le serveur WBEM, à développer les Pilotes WBEM associés à ces modèles, et à les déployer dans les systèmes qui assurent l'instrumentation des équipements du laboratoire. De plus, grâce à l'abstraction des modèles d'expérience, cette couche permet de mutualiser plusieurs laboratoires et de les présenter comme un seul et unique laboratoire virtuel.

Enfin, notre travail rentre dans le cadre de travaux ultérieurs et en cours dédiés au suivi des activités d'apprentissage et à l'exploitation de celles-ci à travers le standard WBEM. Nous avons étendu les résultats de ces travaux pour prendre en compte les traces d'interaction entre apprenants et expériences pour différents objectifs pédagogiques.

12.1.4. Des modèles génériques pour la gestion des expériences et des traces.

Les avantages de l'approche par Modèle adoptée dans notre travail sont multiples :

- Pour les enseignants concepteurs : la possibilité de décrire et manipuler les expériences à l'aide de modèles ouverts et normalisés (et donc portables et partageables), mais également la possibilité d'étendre les modèles proposés pour couvrir des besoins et des aspects plus spécifiques en fonction des objectifs pédagogiques visés, de la nature du public cible, etc.
- Pour les développeurs : la possibilité de construire des outils et IHMs de commande et de supervision fondés sur ces mêmes modèles.
- Pour les enseignants, analystes et chercheurs en EIAH : la possibilité d'effectuer des traitements avancés sur les traces d'activités recueillies sur une expérience (synthèse, classification, rejoue, etc.), mais également le croisement d'autres types d'activités appartenant au même cursus de formation.

Mais le principal avantage du méta-modèle proposé dans ce travail repose sur sa généricité et sa souplesse, qui permettent son extension et sa spécialisation pour les disciplines

scientifiques, technologiques et mathématiques nécessitant des travaux pratiques. Une application au domaine de l'enseignement de l'informatique a été détaillée, dans laquelle nos sous-modèles génériques ont été spécialisés :

- Le méta-modèle dédié à la gestion des laboratoires informatiques en ligne inclut la gestion des ressources ; cette modélisation a été inspirée des travaux existants sur la gestion de l'allocation des ressources dans les grilles de calcul.
- Le méta-modèle dédié à la gestion des objets pédagogiques interactifs « Expérience » composés de ressources offertes par les laboratoires en ligne, augmenté par des propriétés et des associations spécifiques fixées selon la configuration souhaitée par le concepteur pédagogique.
- Le méta-modèle dédié à la gestion des traces d'activités générées par les interfaces de type ligne de commande. Cetype d'interface très répandu au sein des systèmes informatiques existants reste très apprécié des utilisateurs pour sa simplicité, sa souplesse et son efficacité. C'est également un type d'interface offert par la plupart des systèmes informatiques embarqués dans les dispositifs de laboratoires (en particulier les instruments de mesure), ce qui prédispose notre modèle de suivi des activités à être appliqué aux activités sur ces équipements. Bien entendu, des travaux futurs devront s'intéresser au développement de méta-modèles de suivi d'activités prenant place à l'aide d'interfaces plus évoluées (notamment les interfaces graphiques).

12.1.5. Outils et IHMs supports à l'apprentissage et au tutorat

Les outils et IHMs d'apprentissage développés dans le cadre de ce travail sont tous intégrés à la plateforme de télé-enseignement Moodle ; nous ne reprenons pas ici les raisons de ce choix, mais un de nos objectifs était d'offrir tous les outils d'expérimentation à travers l'interface de ce LMS. Pour l'apprenant, ce choix prévient la surcharge cognitive liée à la multiplicité des fenêtres et évite à l'utilisateur l'installation d'un ou de plusieurs logiciels additionnels. Notre implémentation repose sur la technologie AJAX qui ne nécessite même pas l'ajout d'un *plugin* au navigateur, sans sacrifier l'interactivité très élevée requise pour télé-opérer une expérience en ligne. Nous avons validé ces choix par l'IHM que nous avons développée pour exploiter des expériences. Cette interface spécifique, inspirée de la métaphore du *cockpit*, intègre en une seule page web des outils d'exploration et de supervision de l'expérience, des

outils d'ouverture et de gestion de sessions Terminal sur une expérience d'une complexité quelconque, et des outils d'exploitation des traces d'activité.

L'expérimentation collaborative est supportée par deux fonctionnalités distinctes : la lecture en temps réel ou en différé du déroulement d'une session d'un autre apprenant, et l'accès partagé à une même session par plusieurs apprenants.

De la même façon, le tutorat est supporté par les outils d'expérimentation collaborative, conformément au principe socioconstructiviste qui postule qu'un tuteur est un pair avancé. La différence réside dans les privilèges élargis octroyés au tuteur sur les sessions des apprenants ainsi que dans l'utilisation d'outils de supervision plus poussés.

12.2. Perspectives

12.2.1. Perspectives générales

La réussite de notre projet dépend bien sûr de sa maturité et de sa stabilité d'un point de vue technique, mais également de sa généricité et son acceptabilité par la communauté de l'e-formation. Pour le premier point, l'effort de développement doit continuer afin de compléter les composants présentés dans la dernière partie de ce manuscrit, mais également corriger les erreurs signalées par les utilisateurs de l'expérimentation.

Le second point requière le développement d'outils et IHMs légers, accessibles et intuitifs couvrant le plus large spectre des disciplines scientifiques, qui aident à réduire la complexité des tâches de conception pédagogique, de suivi et d'apprentissage dans le cadre de télé-TP en ligne. Les outils et IHMs auteurs au profit des concepteurs de télé-TPs doivent être privilégiés, ainsi que les outils et IHMs de conduite des expériences en ligne, sans oublier le développement et le partage des pilotes WBEM dédiés à l'intégration des dispositifs de laboratoires appartenant à des disciplines différentes.

12.2.1.1. Outils auteurs intuitifs pour la modélisation des expériences

La phase de modélisation CIM des expériences représente une étape clé dans la mise en œuvre d'une activité de télé-TP. Cette tâche peut être réalisée par la plupart des informaticiens, mais elle constitue un blocage pour les concepteurs d'expériences dépourvus

de ces notions de modélisation. Des solutions propriétaires (Rational Rose d'IBM, Visio et WMI Studio de Microsoft) permettent de traduire un modèle CIM exprimé par sa représentation visuelle UML vers son équivalent en langage MOF, mais le langage UML, même s'il est visuel, reste destiné aux informaticiens.

Une première solution consiste à développer une interface graphique intuitive de génération de modèles, fondée sur les technologies du web 2.0 et inspirée des logiciels de modélisation graphiques existants (tels que MS Visio, Open Office/Libre Office Draw, ou DIA). Ces IHMs affichent des palettes d'objets visuels spécifiques qui représentent un dispositif de laboratoire (réel ou simulé) disponible dans (et pris en charge par) un laboratoire distant. Des opérations de type glisser/déposer entre la palette des objets visuels et la fenêtre de travail suffisent à construire une expérience, une palette spécifique pouvant contenir différents types de connexions permettant de créer des liens entre les objets de l'expérience.

De plus, cette IHM de conception devrait permettre au concepteur de créer et de personnaliser ses propres palettes d'objets de laboratoires spécifiques à une discipline non couverte par les palettes existantes. L'outil propriétaire de modélisation graphique LABVIEW ou le module *open source* GMF⁸⁴ de l'environnement de développement Eclipse constituent des briques élémentaires pour atteindre cet objectif.

Une autre solution, immédiate, consiste à encourager le partage des expériences à travers le développement de modèles typiques d'expériences et leur mise à disposition à la communauté via des viviers de connaissances. Il s'agit donc de bien documenter et référencer les modèles d'expériences pour en faciliter la recherche et la réutilisation.

12.2.1.2. Génération automatique d'interfaces de télé-opération

La séparation du modèle d'une expérience et de son IHM a été énoncée comme un avantage de notre approche qui permet d'ouvrir la voie au développement d'outils et IHMs variés et riches pour le contrôle et la supervision d'expériences en ligne, conduits par des équipes de développement indépendantes, poursuivant des objectifs différents et visant des publics variés. Mais ceci peut être perçu comme un handicap pour les enseignants ou institutions qui

⁸⁴ Graphical Modeling Framework

ne possèdent pas les moyens et/ou les ressources humaines nécessaires au métier du génie logiciel.

Une solution consiste à développer un (ou des) logiciel(s) générateur(s) d'interfaces de commande et de supervision d'expériences en ligne, qui produit automatiquement une IHM personnalisée à partir du croisement d'un ou de plusieurs modèles d'IHMs « type » conçues au préalable par des développeurs spécialisés. L'IHM résultante peut être affinée en prenant en compte le profil de l'utilisateur (tuteur ou apprenant, débutant ou utilisateur avancé, etc.) pour n'afficher que les fonctionnalités adéquates à la bonne exécution de l'expérience.

12.2.1.3. Développement rapide de *Pilotes* WBEM

Un autre handicap qui peut entraver le développement des expériences CIM est la nécessité d'implémenter les Pilotes WBEM associés aux dispositifs de laboratoires (s'ils n'existent pas déjà) ; rappelons ici que ces composants représentent des modules logiciels spécifiques aux ressources de laboratoires. Actuellement des environnements de développement facilitent cette tâche, comme SimpleWbem qui a été utilisé dans notre projet ou KonkretCMPI, WBEMServices et WMI CIM Studio (pour les systèmes Microsoft). Toutefois ces outils sont dédiés aux systèmes informatiques, ils ne s'intéressent pas à d'autres types de dispositifs (électriques, chimiques, etc.). Si nous voulons généraliser la mise en ligne de travaux pratiques, des moyens pour simplifier le développement de Pilotes WBEM spécifiques doivent être proposés.

Une solution consiste à élaborer (ou réutiliser) un environnement de programmation visuel (semblable au langage G de LabView, ou PLV de Microsoft) où les Pilotes WBEM résultent de la composition d'objets visuels (qui représentent des objets CIM), alors que le code source est généré et compilé (ou interprété) en arrière plan. Ce genre d'environnement produirait les Pilotes WBEM requis pour opérationnaliser le modèle CIM relatif à un dispositif de laboratoire donné.

Toutefois, le partage des Pilotes WBEM reste le moyen le plus rapide et efficace pour atteindre notre objectif.

12.2.1.4. Intégration effective des expériences dans les scénarios IMS-LD

L'intégration des modèles d'expériences exprimés en langage CIM dans les scénarios IMS-LD représente la prochaine étape clé de ce travail. Rappelons que toutes les conditions sont réunies pour réussir cette intégration : le moniteur d'un scénario IMS-LD (qu'il soit de type A, B ou C) devient un simple client WBEM qui interagit avec les modèles CIM des expériences.

Pour les scénarios de type A, l'expérience est intégrée comme ressource pédagogique à titre descriptif. Pour les scénarios de type B, des conditions peuvent être calculées en utilisant les valeurs des attributs obtenus via les méthodes CIM intrinsèques et/ou les valeurs de retours des méthodes extrinsèques. Pour les scénarios de type C, les événements survenus aux seins des laboratoires (et donc des expériences) peuvent être reçus par le moniteur IMS-LD sous la forme d'instances CIM pour déclencher une réaction telle que l'orientation vers une autre activité, le renvoi vers une ressource, etc.

12.2.1.5. Expérimentations

Nous ne pouvons valider nos propositions qu'après des expérimentations menées dans divers domaines de formation, et à une grande échelle. Cela requière l'instanciation de l'EIAH générique pour une discipline particulière, son déploiement auprès d'un nombre important de groupes d'apprenants, et son évaluation pédagogique et technique.

Dans le cadre de ce travail, une expérimentation pilote a été conduite avec deux groupes d'apprenants appartenant à un public d'ingénieurs en informatique. Un scénario de télé-TP type leur a été proposé, contenant une expérience en réseaux informatiques déployée en deux exemplaires.

Ce test nous a permis de suivre le déroulement du télé-TP par les apprenants-testeurs en utilisant les moyens de suivi intégrés au sein de notre EIAH. Les traces recueillies ainsi que les impressions exprimées par les utilisateurs confortent notre approche : les apprenants ont utilisé aisément et pleinement les outils mis à leur disposition, et en particulier les outils de soutien à l'apprentissage collaboratif (interfaces web de contrôle en mode ligne de commande, rejoue et partage de sessions de travail, exploitation des différents outils de communication, etc.).

Au terme de l'activité pédagogique, les testeurs étaient invités à répondre à un questionnaire en ligne fondé sur les heuristiques de Nielsen et adapté à notre environnement de télé-TP. Les résultats sont encourageants, puisque la majorité des testeurs ont exprimé leur satisfaction vis-à-vis des outils et IHMs mis à leur disposition. Nous avons pu vérifier ce résultat par l'analyse des traces recueillies qui montrent qu'ils se sont familiarisés avec les outils proposés.

Cette expérimentation nous a également permis d'évaluer les faiblesses de l'implémentation actuelle à travers les erreurs signalées par les testeurs, mais aussi par d'autres fonctionnalités souhaitées dans le cadre de genre d'activité. En effet, en plus de l'incomplétude de l'implémentation et de la documentation, certains testeurs ont souhaité disposer de moyens de régulation plus fins du travail d'équipe en ligne, alors que d'autres ont proposé le développement de « Player » de sessions de travail ayant des fonctionnalités de retour en arrière, d'accélération ou de décélération de la vitesse de rejoue.

Malgré les résultats positifs obtenus, nous insistons sur la nécessité d'élargir les tests au plus grand nombre de testeurs appartenant à des institutions et disciplines différents. En particulier, les traces d'activités recueillies par nos outils, conformément au modèle de suivi proposé dans ce travail, doivent être soumises à des experts et chercheurs afin d'y appliquer des techniques d'analyse quantitatives et qualitatives avancées qui nous permettront de confirmer ou infirmer nos propositions, d'aider à améliorer aussi bien nos modèles que nos outils, et bien entendu contribuer aux études sur les usages et comportements des acteurs humains impliqués dans l'e-formation pour les confronter aux théories d'apprentissage.

12.2.2. Perspectives spécifiques au domaine informatique

12.2.2.1. Outils auteurs pour les expériences en informatique

Nous avons proposé deux outils auteurs que le concepteur doit utiliser selon un ordre précis : le premier pour la modélisation des machines virtuelles, le second pour la modélisation des expériences réseau. Afin de faciliter cette tâche de conception et diminuer la charge cognitive liée à l'utilisation de plusieurs outils, un système unique devrait gérer dynamiquement et automatiquement les dépendances entre machines virtuelles et expériences réseau.

Les outils proposés dans ce travail couvrent des expériences en ingénierie informatique qui se focalisent sur l'infrastructure informatique sous-jacente et couvrant par exemple les aspects de la programmation système, des réseaux et des télécommunication. Toutefois ces outils restent valables pour d'autres disciplines informatiques, notamment le génie logiciel et les systèmes d'informations, puisque seul un effort d'extension et de spécialisation relatif à la partie logicielle de notre modèle de base des expériences est nécessaire. Par exemple, des modèles décrivant les environnements de développement (compilateurs, environnements intégrés, etc.) spécifiques au génie logiciel ou ceux relatifs aux bases de données relationnelles doivent être élaborés pour prendre en charge ces disciplines.

12.2.2.2. Gestion intégrée des expériences fondée sur des Pilotes CIM/WBEM natifs

Notre environnement d'expérimentation est actuellement fondé sur un outil tiers, MLN. Malgré la puissance de cet outil, il soulève deux problématiques :

- Son langage de modélisation est trop restreint par rapport aux possibilités offertes par les modèles CIM, sans parler du modèle de référence OVF dédié aux machines virtuelles.
- Pour l'intégrer dans notre EIAH, nous avons développé un Pilote WBEM spécifique qui n'implémente pour le moment que quelques fonctionnalités de base requises pour l'exploitation et la supervision des expériences.

Nous proposons de développer un gestionnaire d'expériences fondé nativement sur l'architecture WBEM et les modèles CIM de virtualisation des systèmes et ressources informatiques. Cette perspective est appuyée par l'initiative VMAN⁸⁵ du DMTF qui vise la gestion de l'intégralité du cycle de vie d'un environnement virtuel à partir d'un client WBEM. Le projet Libvirt⁸⁶, qui implémente les modèles de l'initiative VMAN pour gérer via un client WBEM tout le cycle de vie des machines virtuelles appartenant à huit technologies de virtualisation différentes, constitue un point de départ fort intéressant. Un tel outil permettrait d'enrichir considérablement l'interface de contrôle des expériences pour piloter des composants encore plus fins de l'expérience.

⁸⁵ <http://dmtof.org/standards/vman>

⁸⁶ <http://libvirt.org>

12.2.2.3. Gestion intégrée des expériences dans une plateforme de nuages informatiques

Le développement rapide et la maturité des technologies de virtualisation ont accéléré l'adoption de la technologie des nuages informatiques⁸⁷. A la croisée des chemins entre les grilles de calcul et l'informatique à la demande⁸⁸, elle permet d'externaliser les processus métiers des entreprises vers une infrastructure partagée gérée par une organisation tierce.

L'utilisateur d'un nuage informatique obtient sur demande des capacités de calcul et/ou de stockage, ou plus spécifiquement des services, des applications ou toute autre ressource informatique virtuelle. L'avantage est l'indépendance vis-à-vis des considérations liées aux disponibilités du matériel ou du logiciel, de la puissance de calcul ou de stockage. De plus, les plateformes existantes peuvent être exploitées manuellement via un navigateur web ou automatiquement via des applications qui font appel aux services web exposés par ces plateformes.

Il existe trois formes d'utilisation d'un nuage informatique qui ne sont pas mutuellement exclusives :

- L'infrastructure sur demande, ou IaaS (*Infrastructure as a Service*) : elle permet de louer une infrastructure composée de serveurs, de postes de travail et d'espaces de stockage avec un accès illimité à ces derniers.
- Les plateformes sur demande, ou PaaS (*Platform as a Service*) : elles visent la location d'environnements de développement et de test prêts à l'emploi, éliminant la phase non triviale de recherche du matériel et des logiciels adéquats.
- Les services sur demande, ou SaaS (*Service as a Service*) : ils permettent de louer des applications d'entreprise prêtes à l'emploi comme les ERP, les CRM, etc.

Nous pensons que les nuages informatiques représentent le futur des expériences informatiques à travers ce que nous pouvons appeler « Les expérience sur demande » ou EaaS (*Experiment as a Service*). En effet, le déploiement de nuages privés au sein de chaque université et la mutualisation de ces nuages va permettre d'accroître les possibilités de

⁸⁷ En anglais « Cloud Computing »

⁸⁸ En anglais « Utility Computing »

déploiement d'expériences réalistes. Le modèle IaaS est celui qui convient le mieux à la majorité des expériences informatiques, en particulier celles qui nécessitent le déploiement de réseaux de machines. Le modèle PaaS convient à l'enseignement du génie logiciel où les apprenants peuvent développer des applications sans se soucier de la plateforme et de l'environnement de développement sous-jacent, alors que le modèle SaaS est particulièrement adapté à la collaboration entre enseignants lors de la phase de conception. Finalement, les trois modes conviennent aux activités de conception pédagogique aussi bien des expériences et des télé-TPs que d'autres ressources d'apprentissage.

Chapitre 13. Bibliographie

(ABET, 2009a) Applied Science Accreditation Commission, 2009 Criteria for accrediting Applied Science Programs, disponible en ligne sur <http://www.abet.org/Linked%20Documents-UPDATE/Criteria%20and%20PP/R001%2009-10%20ASAC%20Criteria%208-27-09.pdf>, dernier accès Mai 2010

(ABET, 2009b) Applied Science Accreditation Commission, 2009 Criteria for accrediting Computing Programs, disponible en ligne sur <http://www.abet.org/Linked%20Documents-UPDATE/Criteria%20and%20PP/C001%2010-11%20CAC%20Criteria%2011-16-09.pdf>, dernier accès Mai 2010

(ABET, 2009c) Applied Science Accreditation Commission, 2009 Criteria for accrediting Engineering Programs, disponible en ligne sur <http://www.abet.org/Linked%20Documents-UPDATE/Criteria%20and%20PP/E001%2010-11%20EAC%20Criteria%201-27-10.pdf> , dernier accès Mai 2010

(ABET, 2009d) Science Accreditation Commission, 2009 Criteria for accrediting Technology Science Programs, disponible en ligne sur <http://www.abet.org/Linked%20Documents-UPDATE/Criteria%20and%20PP/T001%2010-11%20TAC%20Criteria%2011-3-09.pdf> , dernier accès Mai 2010

(ADLNet, 2009) Advanced Distributed Learning, Documentation Suite (SCORM 2004 4th Ed.), disponible en ligne sur : <http://www.adlnet.gov/Technologies/scorm/SCORMSDocuments/SCORM%20Resources/Resources.aspx> dernier accès, Septembre 2010.

(Agarwal et Johnston,1998) D.A. Agarwal, S.R. Sachs, W.E. Johnston The reality of collaboratories. *Computer Physics Communications* 110 (1998) 134-141

(Alzand, 2010) Walid Alzand, Instruction design and educational quality, *Procedia Social and Behavioral Sciences* 2 (2010) 4074–4081, 1877-0428 © 2010 Published by Elsevier Ltd.

(Arnaud, 2002) Michel Arnaud, Normes et standards de l'enseignement à distance : enjeux et perspectives. Actes du Colloque TICE 2002, Lyon, 13-15 novembre 2002, Lyon, INSA, p. 57-69.

(Aubé et David, 2003) Aubé, M. & David, R.. Le programme d'adoption du monde de Darwin : une exploitation concrète des TIC selon une approche socio-constructiviste. In Taurisson, A. & Senteni, A.(2003). *Pédagogie.net : L'essor des communautés d'apprentissage*. pp 49-72.

(Baier et al, 2006) Aurelius Baier, Thomas Bernoulli, Torsten Braun, Christoph Graf et Ulrich Ultes-Nitsche. Case Study of the Usage of an Authentication and Autorization Infrastructure (AAI) in an E-Learning Project Information Security South Africa (ISSA 2006), *Proceedings of the ISSA 2006 from Insight to Foresight Conference (ISSA 2006)*, Sandton, South Africa, July 2006.

(Barhms et al, 2003) Paul Barham_, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, Andrew Warfeld , Xen and the art of virtualization, In *Proceedings of the nineteenth ACM symposium on Operating systems principles*, Bolton Landing, NY, USA, Pages: 164 - 177 , 2003

(Baudin et al, 2005) Veronique Baudin, Martin Faust, Hannes Kaufmann, Vivian Litsa, Daisy Mwanza, Arnaud Pierre and Alexandra Totter. The Lab@Future Project “Moving Towards the Future of E-Learning” In. *Technology Enhanced Learning*, Volume 171/2005, 3-18, ISBN978-0-387-24046-6, Springer, 2005

(Baumgartner et al, 2003) Florian Baumgartner, Torsten Braun, Eveline Kurt, Attila Weyland. Virtual Routers: A Tool for Networking Research and Education. *ACM Computer Communication Review*, 33(3):127-135, July 2003.

(Benmohamed, 2006) Benmohamed Hcen. ICTT@LAB: un environnement informatique pour la génération et l'exécution de scénario de télé TP. Thèse en Informatique, Institut National des Sciences Appliqués, Lyon: France, 2006. 279 P.

(Berger et Kam, 1996) Carl Berger et Rosalind Kam. Education 626: Educational Software Design and Authoring. 1996 disponible en ligne sur <http://www.umich.edu/~ed626/define.html> dernier accès Mai 2010

(Berqia et al, 2002) Amine Berqia, Alassane Diop, Jürgen Harms. A Virtual Laboratory for Practical exercises. In Proceedings of International Conference on Engineering Education, Manchester, UK, August 2002.

(Berqia et Diop, 2001) Amine BERQIA - Alassane DIOP - Jüregen HARMS A complete Architecture of a Virtual Telecommunications Laboratory. In Proceedings of International Workshop on Tele-Education in Mechatronics Based on Virtual Laboratories, Weingarten, Germany, July 2001.

(Bilodeau et al, 1999) BilodeauHélène, ProvencherMichelle, Bourdages Louise, André-Jacques , DionneMichel , Gagné Pierre, LebelCéline ,Rada-Donath Alexandro, 1999 Les objectifs pédagogiques dans les activités d'apprentissage de cours universitaires à distance,Revue Distances, Volume 3 Numéro 2 disponible sur le site http://cqfd.telug.quebec.ca/distances/D3_2_d.pdf dernier accès Mai 2010

(Bloom, 1956) Bloom B. S. 1956. Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain. New York: David McKay Co Inc.

(Brasher et Schopmeyer, 2006) Michael Brasher et Karl Schopmeyer, The CIMPLE Provider Development Environment, Presented at the 2006 Management Developers Conference, disponible en ligne sur <http://www.simplewbem.org/CIMPLE-2006.pdf>, dernier accès : Aout, 2010

(Brasher et Schopmeyer, 2007) Michael Brasher et Karl Schopmeyer, CIMPLE and BREVITY, Presented at the 2007 Management Developers Conference, disponible en ligne sur <http://www.simplewbem.org/CIMPLE-2007-MDC.pdf> dernier accès : Aout 2010

(Brasher et Schopmeyer, 2009) Michael Brasher et Karl Schopmeyer, Using CIMPLE V2, A Practical Guide to Developing CIM Providers, disponible en ligne sur http://www.simplewbem.org/Using_CIMPLE2.pdf , dernier accès: Aout 2010

(Broisin, 2006) Broisin Julien. Un Environnement Informatique pour l'Apprentissage Humain au Service de la Virtualisation et de la Gestion des Objets Pédagogiques. Thèse, Informatique, Université Paul Sabatier, Toulouse : France, 2006. 206P

(Bruillard et al, 2000) Éric Bruillard, Élisabeth Delozanne, Pascal Leroux, Paul Delannoy, Xavier Dubourg, Pierre Jacoboni, Jérôme Lehuen, Daniel Luzzati, Philippe Teutsch ,
Quinze ans de recherche informatique sur les sciences et techniques éducatives au LIUM, Sciences et Techniques éducatives Volume 7 n°1 – 2000 pp. 87-145

(Bruner et Hickman, 1983) Bruner J. S. et Hickman M. La conscience, la parole et « la zone proximale » : réflexions sur la théorie de Vygotsky. In. Le développement de l'enfant Savoir faire, savoir dire. Éditions PUF Paris p. 281-292.

(Buitrago, 1999) Gloria CORTES BUITRAGO. Simulations et Contrôle Pédagogique : Architectures Logicielles Réutilisables. Thèse Informatique, l'Université Joseph Fourier - Grenoble I, 1999, 277P.

(Burgos et al, 2005) Daniel Burgos, Michel Arnaud, Patrick Neuhauser et Rob Koper. IMS Learning Design : la flexibilité pédagogique au service des besoins de l'e-formation, In EpiNet la Revue électronique de l'EPI (association Enseignement Public et Informatique) Année 2005, n° 71-80, disponible en ligne sur <http://www.epi.asso.fr/revue/articles/a0512c.htm>, dernier accès juillet 2010.

(BUTOIANU et al. 2010) Butoianu, V., Vidal, P., Verbert, K., Duval, E., Broisin, J., « User context and personalized learning: a federation of Contextualized Attention Metadata », Journal of Universal Computer Science, John Wiley and Sons, Vol. 16 N. 16, p. 2252-2271, 2010.

(BUTOIANU et al. 2011) BUTOIANU V., CATTEAU O., VIDAL P., BROISIN J. (2011). Un Système à Base de Traces pour la Recherche Personnalisée d'Objets Pédagogiques : le cas d'Ariadne Finder. Atelier "Personnalisation de l'apprentissage : quelles approches pour quels besoins ?", EIAH 2011, Mons, Belgique.

(CARRON 2006) Carron, T., Marty, J.C., Heraud, J.M., France, L. (2006). Helping the teacher to re-organize tasks in a collaborative learning activity: an agent-based approach. ICALT 2006, p. 552-554.

(Catteau, 2008) Olivier Catteau. Le Cycle de Vie de l'Objet Pédagogique et de ses Métadonnées. Thèse en Informatique, Université Paul Sabatier, Toulouse : France, 2008, 218 P.

(CBG, 2000) Computing Benchmarking Group "Computing", UK Quality Assurance Agency for Higher Education, 2000

(Clark, 2009) Donald Clark, 2009 Bloom's Taxonomy of Learning Domains, 2009 Disponible en ligne sur <http://www.nwlink.com/~donclark/hrd/bloom.html>, dernier accès Mai 2010

(Clark, 2010) Donald Clark, Instructional System Design <http://www.nwlink.com/~donclark/hrd/sat.html> 2010, dernier accès Mai 2010

(Colwell et al, 2002) Chetz Colwell, Eileen Scanlon, Martyn Cooper. Using remote laboratories to extend access to science and engineering. Computers & Education 38 (2002) 65–76

(Cooper et al, 2002) Martyn Cooper, Alexis Donnelly, Jose Ferreira . Remote Controlled Experiments for Teaching over the internet: A comparison of approaches developed in the PEARL project. In the PEARL Project. 19th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education. 2002 P 119-128

(Cooper et Ferreira, 2009) Martyn Cooper and Jose ´ M.M. Ferreira, Remote Laboratories Extending Access to Science and Engineering Curricular. IEEE TRANSACTIONS ON LEARNING TECHNOLOGIES, VOL. 2, NO. 4, OCTOBER-DECEMBER 2009

(Cordel et al, 2008) Dirk Cordel, Christoph Meinel, Stephan Repp, and Christian Willems. Explorative Learning of Wireless Network Security with Tele-Lab IT-Security. In J. Fong, R. Kwan, and F.L. Wang (Eds.): ICHL 2008, LNCS 5169, pp. 213–224, 2008 Springer

(Corter et al, 2007) James E. Corter, Jeffrey V. Nickerson, Sven K. Esche, and Constantin Chassapis, Seongah IM, Jing Ma. Constructing Reality: A Study of Remote, Hands-On,

and Simulated Laboratories. ACM Transactions on Computer-Human Interaction, Vol. 14, No. 2, Article 7, Publication date: August 2007.

(CUEEP, 2010) Centre Université-Economie d'Education Permanente - Département des Sciences de l'Education, 2010 La pédagogie par objectifs, disponible en ligne, http://cueep.univ-lille1.fr/pedagogie/La_PPO.htm dernier accès: Mai 2010

(Dabbagh, 2005) Pedagogical models for E-Learning: A theory-based design framework. *International Journal of Technology in Teaching and Learning*, 1(1), 25-44.

(Demers et Liull, 1982) Marthe Demers et Georges Liull. L'impérieuse nécessité de l'enseignement des sciences. *Revue des sciences de l'éducation*, vol. 8, n° 1, 1982, p. 91-102. Disponible en ligne sur: <http://id.erudit.org/iderudit/900359ar> dernier accès Mai 2010

(Denning et al, 1989) Peter J. DENNING, Douglas E. COMER, David GRIES, Michael C. MULDER, Allen TUCKER, A. Joe TURNER, and Paul R. YOUNG, "COMPUTING AS A DISCIPLINE" In, *Communications of the ACM*, Volume 32 Number 1. P23. Janvier 1989

(Depover et al, 2006a) Christian Depover, Bruno De Lièvre, Jean-Jacques Quintin, Sandrine Decamps et Filippo Porco. Les modèles d'enseignement et d'apprentissage. Disponible en ligne http://ute.umh.ac.be/site_ute2/menu.htm consulté en avril 2010

(Depover et al, 2006b) Christian Depover, Bruno De Lièvre, Jean-Jacques Quintin, Sandrine Decamps et Filippo Porco Méthodologie de conception des environnements d'apprentissage multimédia. Disponible en ligne http://ute.umh.ac.be/site_ute2/menu.htm, consulté en avril 2010

(Depover et al, 2006c) Christian Depover, Bruno De Lièvre, Jean-Jacques Quintin, Sandrine Decamps et Filippo Porco. Structuration pédagogique d'une formation. Disponible en ligne http://ute.umh.ac.be/site_ute2/menu.htm consulté en avril 2010

(Depover et al, 2006d) Christian Depover, Bruno De Lièvre, Jean-Jacques Quintin, Sandrine Decamps et Filippo Porco. Analyse et la conception des scénarios d'apprentissage. Disponible en ligne, http://ute.umh.ac.be/site_ute2/menu.htm consulté en Mai 2010

(Després et Leroux, 2003) Després Christophe & Leroux Pascal. Le tutorat synchrone en formation à distance : un modèle pour le suivi pédagogique synchrone d'activités d'apprentissage à distance. Environnements Informatiques pour l'Apprentissage Humain 2003, Strasbourg : France (2003) 139-150 (Leroux, 2002) Leroux Pascal. Machines partenaires des apprenants et des enseignants - Étude dans le cadre d'environnements supports de projets pédagogiques, habilitation à diriger des recherches, Le Mans : Université du Maine 2002. 219 P.

(Després, 2001) Christoph Després, Modélisation et conception d'un environnement de suivi pédagogique synchrone d'activités d'apprentissage à distance, Thèse, Informatique, Université du Maine, Le Mans : France, 2001. 286P

(Devis et al, 2003) *Dervis Z. Deniz¹, Atilla Bulancak² and Gökhan Özcan³*. A novel approach to remote laboratories. in 33rd ASEE/IEEE Frontiers in Education Conference November 5-8, 2003

(Dewey, 1922) Dewey J., L'école et l'enfant, Éditions Delachaux et Niestlé, 1922.

(Dillenbourg et al, 1996) Dillenbourg, P., Baker, M., Blaye, A. & O'Malley, C.(1996) The evolution of research on collaborative learning. In E. Spada & P. Reiman (Eds) Learning in Humans and Machine: Towards an interdisciplinary learning science (pp. 189-211). Oxford: Elsevier

(Dillenbourg et al, 2002) Dillenbourg, P., Schneider, D.K. & Synteta, P.Virtual Learning Environments. In A. Dimitracopoulou (Ed). Proceedings of the 3rd Hellenic Conference Information & Communication Technologies in Education" 2002 (pp. 3-18). Kastaniotis Editions, Greece

(Diop et al, 2003) Alassane DIOP, Amine BERQIA et Jürgen HARMS. Remote Real Laboratory: Linux installation and configuration In Proceedings of International Conference on Internet Computing, Las Vegas, NV, USA, June 2003.

(DMTF, 2003) Site web du projet WBEM Service. Disponible en ligne sur <http://wbemservices.sourceforge.net/>, dernier accès Juin 2010

(DMTF, 2003b) DMTF “CIM Network Model White Paper, 2003, disponible en ligne sur ”http://www.dmtf.org/standards/published_documents/DSP015.pdf dernier accès: Aout 2010

(DMTF, 2003d) Distributed Management Task Force. CIM User and Security Model White Paper. Rapport technique, DSP0139, 2003, 11 p. Disponible sur internet : <http://www.dmtf.org/standards/documents/CIM/DSP0139.pdf> , dernier accès en juin 2010.

(DMTF, 2006a) DMTF “Computer System Profile” disponible en ligne sur http://www.dmtf.org/standards/published_documents/DSP1052.pdf dernier accès: Aout 2010

(DMTF, 2006b) DMTF “IP Interface Profile” disponible en ligne sur http://www.dmtf.org/standards/published_documents/DSP1036.pdf dernier accès: Aout 2010

(DMTF, 2006c) DMTF “Host LAN Network Port Profile” disponible en ligne sur http://www.dmtf.org/standards/published_documents/DSP0816.pdf dernier accès: Aout 2010

(DMTF, 2007) Distributed Management Task Force. CIM Query Language Specification. Rapport technique, DSP0202, 2007, 68p. Disponible sur internet : http://www.dmtf.org/standards/published_documents/DSP0202.pdf dernier accès juin 2010

(DMTF, 2007a) DMTF “CIM System Virtualization Model White Paper” disponible en ligne sur http://www.dmtf.org/standards/published_documents/DSP2013.pdf

(DMTF, 2007b) DMTF « System Virtualization Profile » disponible en ligne sur http://www.dmtf.org/standards/published_documents/DSP1042.pdf dernier accès: Aout 2010

(DMTF, 2009) DMTF, « Resource Allocation Profile » 2009, disponible en ligne sur, http://www.dmtf.org/standards/published_documents/DSP1041_1.1.0.pdf dernier accès [juin 2010](#)

(DMTF, 2009b) http://www.dmtf.org/standards/cim/cim_schema_v220/CIM_User.pdf, 2009.

(DMTF, 2010a) DMTF, « CIM Schema: Version 2.25.0 » disponible en ligne sur l'URL ; http://www.dmtf.org/standards/cim/cim_schema_v2250/cim_schema_2.25.0Final-Doc.zip, dernier accès juin 2010.

(DMTF, 2010b) DMTF, « CLP Service Profile » disponible en ligne sur l'URL http://www.dmtf.org/standards/published_documents/DSP1005_1.0.1.pdf, dernier accès : juin 2010

(Doise et Mugny, 1981) Doise W., Mugny G., Le développement social de l'intelligence InterEditions, Paris, 1981.

(Donnette et Hannequin, 2007) Benoît Donnette et David Hannequin, La virtualisation, livre blanc, disponible en ligne sur : https://www.08000linux.com/public//page_attachments/0000/0001/Virtualisation_Linagoragroup_WB_V1.0.pdf, dernier accès juillet 2010

(Dourish et Bellotti, 1992) Paul Dourish and Victoria Bellotti. Awareness and Coordination in Shared Workspaces. CSCW Proceedings 1992

(Engeström, 1987) Engeström Yrjö, "Learning by Expanding: An Activity-Theoretical Approach to Developmental Research." Helsinki: Orienta-Konsultit Oy, Finland. 1987

(Faerber, 2001a) Faerber Richard, Accompagner les apprentissages à distance et collaborer en petits groupes. Quatrième congrès AECSE 2001, Villeneuve d'Ascq, 5, 6, 7 et 8 septembre 2001, disponible en ligne sur : http://faerber.u-strasbg.fr/publi/aecse_faerber.PDF , dernier accès Mai 2010

(Faerber, 2001b) Faerber Richard, Un espace métaphorique pour une formation à distance, Les communautés en ligne, Dossier de l'ingénierie éducative n°36 Octobre 2001, CNDP, pages 62-65

(Faerber, 2001c) Faerber Richard, Une métaphore spatiale et des outils intégrés pour des apprentissages coopératifs à distance : ACOLAD actes du colloque JRES 2001 Lyon, 10 - 15 décembre 2001 p. 197-204

(Faerber, 2003a) Faerber Richard. Groupements, processus pédagogiques et quelques contraintes liés à un environnement virtuel d'apprentissage. In : Desmoulins, C, Marquet, P. & Bouhineau, D. (Eds). Environnements Informatiques pour l'apprentissage humain, Avril 2003,. Strasbourg, France.

(Fareber, 2003b) Faerber Richard. Présentation de la notion de collecticiel. 2003 Disponible en ligne sur <http://faerber.u-strasbg.fr/collecticiels/collecticiels.html>, dernier accès Mai 2010

(Fareber, 2004) Richard Fareber, Caractérisation des situations d'apprentissage en groupe, Revue STICEF, Volume 11,2004, ISSN : 1764-7223, mis en ligne le 30/12/2004, <http://sticef.org>

(Ferraris et a, 2007) Ferraris, C., Martel, C., Vignollet, L.: LDL for Collaborative Activities, in Botturi, L. & Stubbs, T. (eds) Handbook of Visual Languages in Instructional Design: Theories and Practices. Hershey, PA: Idea Group, 2007.

(Ferraris, et al, 2005) Ferraris C., Lejeune A., Vignollet L., David J.-P Modélisation de scénarios pédagogiques collaboratifs. Dans Actes de la conférence EIAH 2005, disponible en ligne sur : <http://hal.archives-ouvertes.fr/hal-00005687/fr/>

(Ferreira et al, 2002) Jose Martins Ferreira, Ricardo Costa, Gustavo Alves and Martyn Cooper. The PEARL Digital Electronics Lab: Full access to the workbench via the web at the 13th Conference on Innovations in Education for Electrical and Information Engineering EAEEIE, York, 2002

(Festor et Benyoussef, 2000) Olivier Festor et Nizar Ben Youssef. WBEM, rapport de recherche N° 3927, 21 Avril 2000, accessible en ligne sur l'URL : <http://hal.inria.fr/docs/00/07/27/25/PDF/RR-3927.pdf> P. 75 dernier accès, juin 2010

(Fiuczynski, 2006) Marc E. Fiuczynski PlanetLab: Overview, History, and Future Directions, ACM SIGOPS Operating Systems Review, Volume 40 , Issue 1 (January 2006) Pages: 6 - 10 (2006) ISSN:0163-5980

(FORMID, 2010) site web du projet FORMID, disponible en ligne sur <http://www-clips.imag.fr/arcade/projets/FORMID/FORMID.html>, dernier accès, septembre 2010

(Garcia-Zubia et al, 2007) Javier Garcia-Zubia, Pablo Orduña, Diego Lopez-de-Ipina, Unai Hernandez, Ivan Trueba, Remote Laboratories from the software Point of View, In Luis Gomes and Javier García-Zubia(eds). Advances on remote laboratories and e-learning experiences, 131-149 ISBN: 978-84-9830-077-2 Bilbao, 2007

(George et Leroux, 2002) Sébastien George et Pascal Leroux, Un environnement pédagogique dédié aux projets collectifs à distance intégrant une assistance aux apprenants et au chef de projet, 3ème conférence internationale Technologies de l'Information et de la Communication dans l'Enseignement d'Ingénieurs et dans l'industrie (TICE 2002), Lyon, France, 13-15 octobre, 2002, p. 289-295

(George, 2001) Sébastien George, Apprentissage collectif à distance, SPLACH : un environnement informatique support d'une pédagogie de projet, Thèse, Informatique, Université du Maine, Le Mans: France, 2001. 356P

(Gillet et al, 2003) D.Gillet, F.Geoffroy, K. Zeramdini, A. V. Nguyen, Y.Rekik, and Y. Piguet, "The cockpit: an effective metaphor for web-based experimentation in engineering education," International Journal of Engineering Education, vol.19, no.3, pp. 389–397, 2003

(Gravier et al, 2008) C. Gravier¹, J. Fayolle¹, B. Bayard¹, M. Ates¹ and J. Lardon¹. State of the art about remote laboratories paradigms - foundations of ongoing mutations. International Journal of Online Engineering 4, 1 (2008) <http://www.online-journals.org/index.php/i-joe/article/view/480/391>

(Gravier, 2007) Gravier Christoph. Vers la généralisation de manipulations distantes et collaboratives d'instruments de haute technologie. Thèse Informatique, Université Jean Monnet de Saint-Etienne : France, 2007, 238 P.

(Gross et Wolfgang, 2003) Tom Gross, Wolfgang Prinz, Awareness in context: a light-weight approach, Proceedings of the eighth conference on European Conference on Computer Supported Cooperative Work, Helsinki, Finland ,Pages: 295 - 314 , 2003

(Guéraud et al, 2004) Viviane Guéraud, Jean-Michel Adam, Jean-Philippe Pernin, Gaëlle Calvary, Jean-Pierre David. L'exploitation d'Objets Pédagogiques Interactifs à distance :

le projet FORMID. Revue STICEF, Volume 11, 2004, ISSN : 1764-7223, mis en ligne le 25/05/2004, <http://sticef.org>

(Guéraud et al, 2007) Viviane Guéraud, Sandra Michelet, Jean-Michel Adam. Suivi de classe à distance : propositions génériques et expérimentation en électricité. Environnements Informatiques d'Apprentissage Humain, Université de Lausanne, Suisse 27-29 Juin 2007

(Guéraud et Cagnat, 2004) Viviane Guéraud et Jean-Michel Cagnat. Suivi à distance de classe virtuelle active *Technologies de l'Information et de la Connaissance dans l'Enseignement Supérieur et de l'Industrie*, Compiègne : France (2004)

(Guéraud, 2005) Viviane GUERAUD Approche Auteur pour les Situations Actives d'Apprentissage: Scénarios, Suivi et Ingénierie. Habilitation à diriger des recherches, l'Université Joseph Fourier – Grenoble 1, 2005, 136 P.

(Gustavo et al, 2007) Gustavo R. Alves, Manuel G. Gericota, Juarez B. Silva, João Bosco Alves, Large and small scale networks of remote labs : a survey .In Luis Gomes and Javier García-Zubia 2007. Advances on remote laboratories and e-learning experiences, 15-34 , ISBN: 978-84-9830-077-2 Bilbao, 2007

(Henri et Basque, 2003) France Henri et Jausianne Basque, Conception d'activités d'apprentissage collaboratif en mode virtuel. In Deaudlin Colette et Nault Thérèse. Collaborer pour apprendre et faire apprendre, la place des technologies, 2003 Presse de l'université du Quebec, Collection Education Recherche ISBN 2-7605-1228-2 P. 29-53

(Hernández-Leo et al, 2004) Hernández-Leo D., Asensio-Pérez J.I., Dimitriadis Y.A., « IMS Learning Design Support for the Formalization of Collaborative Learning Patterns », Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT'04), 30 August - 1 September, P. 350-354 2004

(Hmelo-Silver et al, 2006) Cindy E. Hmelo-Silver, Ravit Golan Duncan, and Clark A. Chinn. Scaffolding and Achievement in Problem-Based and Inquiry Learning: A Response to Kirschner, Sweller, and Clark. 2006 EDUCATIONAL PSYCHOLOGIST, 42(2), 99–107

(Hobbs, 2004) Chris Hobbs. A practical Approach to WBEM/CIM Management, Auerbach Publications © 2004 (325 pages) ISBN: 0849323061

(Hoffmann, 2005) Hans Hoffmann Grid Computing, E-Science and Applications in industry in Wirtschaftsinformatik 2005 eEconomy, eGovernment, eSociety(2005) ISBN 978-3-7908-1574-0 (Print) 978-3-7908-1624-2 (Online) P. 1715-1731

(Hofstein et Lunetta, 2003) Hofstein Avi et Lunetta Vincent N. 2003. The laboratory in science education: Foundations for the Twenty-First Century. 2003 Wiley Periodicals, Inc. *Sci Ed* **88**:28 – 54, 2004; published online in Wiley InterScience (www.interscience.wiley.com). DOI 10.1002/sce.10106

(Hu et al, 2004) Ji Hu, Christoph Meinel et Michael Schmitt Tele-Lab IT Security: An Architecture for Interactive Lessons for Security Education. SIGCSE'04, March 3–7, 2004, Norfolk, Virginia, USA. ACM

(Hu et al, 2004a) Ji Hu, Dirk Cordel, Christoph Meinel A Virtual Laboratory for IT Security Education In Proc. EMISA 2004 Luxembourg, 2004, pp. 60-71

(Hu et al, 2004b) Ji Hu, Christoph Meinel, Michael Schmitt Tele-Lab IT Security: An Architecture for Interactive Lessons for Security Education In Proc. ACM SIGCSE 2004 Norfolk (Virginia, USA), 2004, pp. 412-416

(Hu et al, 2005) Ji Hu, Dirk Cordel, Christoph Meinel Virtual Machine Management for Tele-Lab "IT-Security" Server In Proc. *IEEE ISCC 2005* La Manga del Mar Menor, (Cartagena, Spain), 2005

(Hu et al, 2006) Ji Hu, Dirk Cordel, and Christoph Meinel New Media for Teaching Applied Cryptography and Network Security. In W. Nejdl and K. Tochtermann (Eds.): EC-TEL 2006, LNCS 4227, pp. 488–493, 2006. Springer

(Hu et Meinel, 2004) Ji Hu, Christoph Meinel Tele-Lab IT-Security: A Means to Build Security Laboratories on the Web Proc. IEEE AINA 2004 Fukuoka (Japan), 2004, pp. 285-288

(IEEE, 2003) IEEE-SA Standards Board, 1484.1TM IEEE Standard for Learning Technology—Learning Technology Systems Architecture (LTSA), Approved 12 June 2003, PRINT ISBN 0-7381-3714-6 SH95141, PDF ISBN 0-7381-3715-4 SS95141

(IEEE, 2003) IEEE-SA Standards Board, 1484.1TM IEEE Standard for Learning Technology—Learning Technology Systems Architecture (LTSA), Approved 12 June 2003, PRINT ISBN 0-7381-3714-6 SH95141, PDF ISBN 0-7381-3715-4 SS95141

(IEEE, 2009) IEEE-SA Standards Board , 1484.12.1TM IEEE Standard for Learning Object Metadata, Approved 13 June 2002, Reaffirmed 13 May 2009, Print: ISBN 0-7381-3297-7 SH95001, PDF: ISBN 0-7381-3298-5 SS95001.

(IMS Global, 2007) IMS Global Learning Consortium, IMS Content Packaging Information Model, disponible en ligne sur http://www.imsglobal.org/content/packaging/cpv1p2pd2/imscp_infov1p2pd2.html, dernier accès: juillet 2010

(IMSGlobal, 2003) IMS Global Learning Consortium, Learning Design Specification. 2003, disponible en ligne sur :<http://www.imsglobal.org/learningdesign/>

(Jara et al, 2009) Carlos A. Jara, Francisco A. Candelas, Fernando Torres, Sebastian Dormido, Francisco Esquembre, Oscar Reinoso. Real-time collaboration of virtual laboratories through the Internet. *Computers & Education* 52 (2009) 126–140

(JEAN-DAUBIAS 2005) Jean-Daubias, S., Eyssautier-Bavay, C. (2005). An environment helping teachers to track students' competencies. *AIED* 2005, p. 19-23.

(Joanna, 1996) Allan, Joanna, Learning outcomes in higher education. *Studies in Higher Education*, 03075079, Mar96, Vol. 21, Issue 1

(Johnston, 2002) William E. Johnston. Computational and data Grids in large-scale science and engineering. *Future Generation Computer Systems* 18 (2002) P.1085–1100

(Jones, 2005) Bob Jones. An Overview of the EGEE Project. *In Peer-to-Peer, Grid, and Service-Oriented in Digital Library Architectures*. 2005 ISBN 978-3-540-28711-7 P. 1-8

(Jones, 2006) Tim Jones, Virtual Linux, An overview of virtualization methods, architectures, and implementations, dernière mise à jour: décembre 2006, disponible en ligne sur : <http://www.ibm.com/developerworks/library/l-linuxvirt/> , dernier accès : juillet 2010

(JTFCC, 2004) The Joint Task Force on Computing Curricula IEEE Computer Society Association for Computing Machinery “Computer Engineering 2004 Curriculum Guidelines for Undergraduate degree Programs in Computer Engineering” A Report in the Computing Curricula Series, 2004

(JTFCC, 2005) The Joint Task Force for Computing Curricula “Computing Curricula 2005 The Overview Report covering undergraduate degree programs in Computer Engineering, Computer Science, Information Systems, Information Technology, Software Engineering”, September 2005

(Karsenti et Fortin, 2003) Thierry Karsenti et Toussain Fortin, Collaboration par les TIC, nouveau défi de la formation pratique ? In Deaudlin Colette et Nault Thérèse. Collaborer pour apprendre et faire apprendre, la place des technologies, 2003 Presse de l’université du Québec, Collection Education Recherche ISBN 2-7605-1228-2 P. 83-101

(Kirschner, 1992) Paul A. Kirschner, Epistemology, Practical Work and Academic Skills in Science Education, Science & Education 1, 273-299, 1992

(Koper et al, 2003) Koper R., Olivier B. & Anderson T., "*IMS Learning Design Information Model*", Eds(2003), IMS Global Learning Consortium Inc., version 1.0, 20 Janvier 2003. Disponible en ligne sur :<http://www.msglobal.org/learningdesign/>, dernier accès juillet 2010

(LANGDALE, 2010) LANGDALE Consultant. « CIMTool : Bridge from CIM to Applications ». Disponible en ligne sur <http://files.cimtool.org/CIMUGPresentation.pdf>, dernier accès, juin 2010

(Lelevé *et al*, 2002) Arnaud Lelevé, Cécile Meyer, Patrick Prevot. Télé-TP : Premiers pas vers une modélisation. In Technologies de l'Information et de la Communication dans les Enseignements d'ingénieurs et dans l'industrie, Villeurbanne: France, 2002 disponible en ligne sur l’URL : <http://edutice.archives-ouvertes.fr/docs/00/02/70/12/PDF/leleve.pdf>

(Lelvé et al 2003) (Lelvé et al 2003)Arnaud, Hcene Benmohamed, Patrick Prevot et Cécile Meryer. Remote Laboratory Towards an integrated training system

(Lelvé et al, 2004) Arnaud Lelvé, Patrick Prevot, Hcene Benmohamed et Mehdi Benadi. Generic E-LAB platforms and Elearning standards. Proceedings of International

Conference on Computer Aided Learning in Engineering Education (CALIE 2004), Grenoble, France, p.0 (2004/02/18)

(Leroux, 2002) Leroux Pascal, Machines partenaires des apprenants et des enseignants - Étude dans le cadre d'environnements supports de projets pédagogiques, habilitation à diriger des recherches, Le Mans : Université du Maine 2002. 219 P.

(LIP 09) <http://www.imsglobal.org/profiles/lipbest01.html>, 2009.

(MA et Nickerson, 2006) Jing M.A. et Jeffrey V. Nickerson. Hands-On, Simulated, and Remote Laboratories: A Comparative Literature Review, ACM Computing Surveys, Vol. 38, No. 3, Article 7, Publication date: September 2006.

(Mager, 1962) Mager R. Preparing Instructional Objectives, revised 2nd edition (Belmont, CA, David Lake Publishers).

(Markopoulos et De Ruyter, 2009) Panos Markopoulos, Boris De Ruyter and Wendy Mackay .Awareness Systems Advances in Theory, Methodology and Design, 2009 , ISBN978-1-84882-476-8 (Print) 978-1-84882-477-5 (Online)

(Martel et al, 2007) Martel C., Lejeune A., Ferraris C., Vignollet L. Scénariser les 4 piliers de la pédagogie. Dans Actes de la conférence EIAH 2007 - (2007) disponible en ligne sur <http://hal.archives-ouvertes.fr/hal-00161454/fr>

(Marthie et Kirshcner, 1995) Marthie A. M. Meester and Paul A. Kirschner, 1995. Practical Work at the Open University of the Netherlands. Journal of Science Education and Technolo~, VoL 4, No. 2, 1995

(MARTINEZ 1989) Martinez, M.E., Lippon (1989), Assessment for learning, Education Leadership, 46, p. 73-75.

(Meinel et al, 2003) Christoph Meinel, Ji Hu, Michael Schmitt, Christian Willems A Tutoring System for IT-Security Proc. IFIP WISE 2003, Monterey (California, USA), 2003, pp. 51-60

(Mergel, 1998) Brenda Mergel. Instructional Design & Learning Theory. 1998, Disponible en ligne sur

<http://www.usask.ca/education/coursework/802papers/mergel/brenda.htm>, dernier accès Mai 2010.

(Millar, 2004) Millar Robin, 2004. The role of practical work in the teaching and learning of science. Paper prepared for the Committee: High School Science Laboratories: Role and Vision, National Academy of Sciences, Washington, DC Disponible en ligne: http://www7.nationalacademies.org/bose/Robin_Millar_Presentation_Jun_04.pdf

(Min-Kyung et Hee-Cheol, 2007) Min-Kyung Kim and Hee-Cheol Kim, How to Build Awareness-Supported Systems Without Sacrificing Privacy. W. Shen et al. (Eds.): CSCWD 2006, LNCS 4402, pp. 609–618, 2007. Springer-Verlag Berlin Heidelberg 2007

(Moodle, 2009) <http://docs.moodle.org/en/Philosophy>, dernier accès Aout, 2010

(Moodle, 2010a) http://docs.moodle.org/en/About_Moodle, dernier accès Aout 2010

(Moodle, 2010b) <http://docs.moodle.org/en/Pedagogy>, dernier accès Aout 2010

(Müller et Erb, 2007) Dieter Müller, Heinz-H. Erbe, Collaborative Remote Laboratories in Engineering Education: Challenges and Visions. In Luis Gomes and Javier García-Zubia(eds). Advances on remote laboratories and e-learning experiences, 35-59 ISBN: 978-84-9830-077-2 Bilbao, 2007

(Murray, 1999) Murray T., « Authoring Intelligent Tutoring Systems: An Analysis of the State of the Art », International Journal of Artificial Intelligence in Education, 1999, Vol. 10, pp 98-129.

(Mwanza, 2002) Mwanza Daizy “Towards an Activity-Oriented Design Method for HCI Research and Practice.” PhD Thesis - The Open University, United Kingdom. 2002

(Nedic et al, 2003) Zorica Nedic, Jan Machotka and Andrew Nafalski. Remote laboratories versus virtual and real laboratories. in 33rd ASEE/IEEE Frontiers in Education Conference November 5-8, 2003

(NetCur, 2000) National Work Forge Center for Emerging Technologies “College curriculum: Networking Curriculum”, 2000

(Nickerson et al, 2007) Jeffrey V. Nickerson , James E. Corter , Sven K. Esche , Constantin Chassapis. A model for evaluating the effectiveness of remote engineering laboratories and simulations in education. Computers & Education 49 (2007) 708–725

(Owezarski et al, 2008) Philippe Owezarski, Pascal Berthou, Yann Labit, David Gauchard, LaasNetExp: a generic polymorphic platform for network emulation and experiments. Proceedings of the 4th International Conference on Testbeds and research infrastructures for the development of networks & communities, 2008

(Papert, 1981) Papert Seymour. Jaillissement de l'esprit, Éditions Flammarion. 1981

(Paquette et al, 1997) G. Paquette, F.Crevier, C. Aubin. Méthode d'ingénierie d'un système d'apprentissage (MISA). Revue Informations In Cognito, numéro 8, 1997. disponible en ligne sur : <http://www.licef.telug.ugam.ca/Portals/29/docs/pub/ingenierie/misa.doc> dernier accès Mai 2010

(Patil et Kobsa , 2009) Sameer Patil and Alfred Kobsa, Privacy Considerations in Awareness Systems: Designing with Privacy in Mind. In Panos Markopoulos, Boris De Ruyter and Wendy Mackay .Awareness Systems Advances in Theory, Methodology and Design, 2009, 187-206, ISBN978-1-84882-476-8 (Print) 978-1-84882-477-5 (Online)

(Pearl, 2000D2.1) PEARL Consortium. Project IST-1999-12550, Review of Foundation Technologies, Technical and Standards Issues. URL: <http://iet.open.ac.uk/pearl/deliverables/deliver.htm> dernier accès janvier 2007

(Pearl, 2000D3.1) PEARL Consortium. Project IST-1999-12550, Review of Pedagogy underlying the PEARL approach. URL: <http://iet.open.ac.uk/pearl/deliverables/deliver.htm> dernier accès janvier 2007

(Pearl, 2000D3.2) PEARL Consortium. Project IST-1999-12550, End-users' needs report

(Pearl, 2000D3.3) PEARL Consortium. Project IST-1999-12550, Curriculum review and institutional issues. URL: <http://iet.open.ac.uk/pearl/deliverables/deliver.htm> dernier accès janvier 2007

(Pearl, 2000D3.4) PEARL Consortium. Project IST-1999-12550, Detailed description of experimental activities that will be implemented for validation in the project. URL: <http://iet.open.ac.uk/pearl/deliverables/deliver.htm> dernier accès janvier 2007

(Pearl, 2000D4.1) PEARL Consortium. Project IST-1999-12550, Functional Specification for PEARL system. URL: <http://iet.open.ac.uk/pearl/deliverables/deliver.htm> dernier accès janvier 2007

(Pearl, 2001D4.2) PEARL Consortium. Project IST-1999-12550, Technical Specification for PEARL system and component modules. URL: <http://iet.open.ac.uk/pearl/deliverables/deliver.htm> dernier accès janvier 2007

(Pearl, 2001D5.1) PEARL Consortium. Project IST-1999-12550, System Wide Design. URL: <http://iet.open.ac.uk/pearl/deliverables/deliver.htm> dernier accès janvier 2007

(Pearl, 2001D7.1) PEARL Consortium. Project IST-1999-12550, PEARL Practical Templates and Tutor Guidelines (Issue 1) URL: <http://iet.open.ac.uk/pearl/deliverables/deliver.htm> dernier accès janvier 2007

(Pearl, 2001D8.1) PEARL Consortium. Project IST-1999-12550, User Interface Meta Layer. URL: <http://iet.open.ac.uk/pearl/deliverables/deliver.htm> dernier accès janvier 2007

(Pearl, 2002D5.3) PEARL Consortium. Project IST-1999-12550, Interim Prototypes (for initial validation phase) URL: <http://iet.open.ac.uk/pearl/deliverables/deliver.htm> dernier accès janvier 2007

(Pearl, 2002D8.2) PEARL Consortium. Project IST-1999-12550, Interim Prototype User Interfaces. URL: <http://iet.open.ac.uk/pearl/deliverables/deliver.htm> dernier accès janvier 2007

(Peraya, 2001) D. Peraya, *STAF 17, La formation à distance. Un cadre de référence, Version 01.01, 2001*, Disponible en ligne sur http://tecfa.unige.ch/tecfa/teaching/staf17/0102/ress/doc/p1_fad/fadcadre.pdf

(Pernin et Lejeune, 2004) Jean-Philippe PERNIN, Anne LEJEUNE, Dispositifs d'apprentissage instrumentés par les technologies : vers une ingénierie centrée sur les

scénarios, Technologies de l'Information et de la Connaissance dans l'Enseignement Supérieur et de l'Industrie, Compiègne : France (2004)

(Pernin, 1996) Jean-Philippe Pernin M.A.R.S. Un modèle opérationnel de conception de simulations pédagogiques. Thèse Informatique, l'Université Joseph Fourier - Grenoble I, 1996, 276 P.

(Pernin, 2003) Jean-Philippe PERNIN, Quels modèles et quels outils pour la scénarisation d'activités dans les nouveaux dispositifs d'apprentissage ?, Séminaire "TIC, nouveaux métiers et nouveaux dispositifs d'apprentissage", 2003

(Quintin et al, 2005) Quintin, J.-J. et Depover, C. et Degache, C. Le rôle du scénario pédagogique dans l'analyse d'une formation à distance. Analyse d'un scénario pédagogique à partir d'éléments de caractérisation définis. Le cas de la formation Galanet. EIAH 2005, Mai 2005, Montpellier, France.

(Quintin et Depover, 2008) Jean-Jacques Quintin & Christian Depover, « Design pédagogique d'un environnement de formation à distance », Lidil, 28 | 2003, [En ligne], mis en ligne le 15 avril 2008. URL : <http://lidil.revues.org/index1603.html>. Consulté le 21 septembre 2010.

(RAMANDALAHY et al. 09) Ramandalahy, T., Vidal, P., Broisin, J., « Opening Learner Profiles across Heterogeneous Applications », IEEE International Conference on Advanced Learning Technologies, ICALT 2009, Riga, Lettonie, 14-18 Juillet 2009.

(Rimondini et al, 2007) Massimo Rimondini. Emulation of Computer Networks with Netkit. Technical Report RT-DIA-113-2007

(Rittenbruch et al, 2009) Markus Rittenbruch, Tim Mansfield, and Stephen Viller. Design and Evaluation of Intentionally Enriched Awareness. In Panos Markopoulos, Boris De Ruyter and Wendy Mackay .Awareness Systems Advances in Theory, Methodology and Design, 2009, 367-395, ISBN978-1-84882-476-8 (Print) 978-1-84882-477-5 (Online)

(Rittenbruch et McEwan ,2009) Markus Rittenbruch and Gregor McEwan. An Historical Reflection of Awareness in Collaboration. In Panos Markopoulos, Boris De Ruyter and Wendy Mackay .Awareness Systems Advances in Theory, Methodology and Design, 2009, 3-48, ISBN978-1-84882-476-8 (Print) 978-1-84882-477-5 (Online)

(RoboTeach, 2010) site web du projet RoboTeach, disponible en ligne sur http://perso.univ-lemans.fr/~e093351/RP/existant/fiche_roboteach.php3.htm , dernier accès: juin 2010

(Rocard et al, 2007) Michel Rocard (président), Peter Csermely, Doris Jorde, Dieter Lenzen, Harriet Walberg-Henriksson, Valerie Hemmo 2008. L'enseignement scientifique aujourd'hui : une pédagogie renouvelé pour l'avenir de l'europe. Rapport du Groupe de haut niveau sur l'enseignement scientifique ISBN – 978-92-79-05660-4 ISSN 1018-5593. Disponible en ligne: http://ec.europa.eu/research/science-society/document_library/pdf_06/report-rocard-on-science-education_fr.pdf

(Romney et Stevenson , 2004) Gordon W. Romney et Brady R. Stevenson An Isolated, Multi-platform Network Sandbox for Teaching IT Security System Engineers. SIGITE'04, October 28-30, 2004, Salt Lake City, UT USA. ACM

(Salmi et Jaillet, 2005) SALMI Louiza et JAILLET Alain, « Pertinence des normes et standards dans les dispositifs de formation à distance, Congrès », L'information numérique et les enjeux de la société de l'information. Colloque international, Tunis, TUNISIE (14/04/2005) 2005 (1), n° 13-15 (586 p.) [Document : 21 p.] (1p.1/4), [Notes: Volume 1 sur 3], pp. 489 509

(Salus, 1994) Peter H. Salus. *A Quarter-Century of Unix*. Addison-Wesley. 1994. ISBN 0-201-54777-5.

(Scanlon et al, 2004) Eileen Scanlon, Chetz Colwell , Martyn Cooper, Terry Di Paolo. Remote experiments, re-versioning and re-thinking science learning. *Computers & Education* 43 (2004) 153–163

(Schaf et al, 2009) F.M. Schaf a,*, D. Müller b, F.W. Bruns b, C.E. Pereira a, H.-H. Erbe, Collaborative learning and engineering workspaces, *Annual Reviews in Control* 33 (2009) 246–252

(SCORE, 2008) Science Community Representing Education 2008. Practical Work in Science : a report and proposal for strategic framework. Disponible en ligne: http://www.score-education.org/downloads/practical_work/report.pdf

(SCTIC, 2002) Groupe de travail sur les normes et standards de la formation en ligne du Sous-comité sur les technologies de l'information et de la communication (SCTIC) Les normes et standards de la formation en ligne, états des lieux et enjeux, 2004, disponible en ligne sur : <http://www.crepuq.qc.ca/IMG/pdf/norm-0210-d-rapport.pdf> dernier accès : Mai 2010

(Sheridan, 1992) Sheridan, T. B. Musings on telepresence and virtual presence. Presence: Teleoper. Virtual Environ.1, 120–125. 1992

(Sheridan, 1999) Sheridan, T. B. Descartes, Heidegger, Gibson, and God: Towards an eclectic ontology of presence. Presence: Teleoper. Virtual Environ. 8, 5, 551–559. 1999

(Singer et al, 2005) Susan R. Singer, Margaret L. Hilton, et Heidi A. Schweingruber, Editors, America's Lab Report: Investigations in High School Science, Committee on High School Science Laboratories: Role and Vision, National Research Council, ISBN: 0-309-65286-3, 254 P, 7 x 10, (2005)

(Smile, 2010) Equipe Administration Système Smile, Livre Blanc Virtualisation, disponible en ligne sur <http://www.smile.fr/Livres-blancs/Systeme-et-infrastructure/La-virtualisation-open-source>, dernier accès : juillet 2010

(Sonnenwald et al, 2004) Diane H. Sonnenwald , Kelly L. Maglaughlin , Mary C. Whitton Designing to support situation awareness across distances: an example from a scientific collaboratory. Information Processing and Management 40 (2004) 989–1011

(Steinemann et al, 2002) Marc-Alain Steinemann, Stefan Zimmerli, Thomas Jampen, Torsten Braun Global Architecture and Partial Prototype Implementation for Enhanced Remote Courses. In Proceedings of IASTED International Conference on Computers and Advanced Technology in Education, Cancun, Mexico, May 2002.

(Steinemann et al, 2002) Marc-Alain Steinemann, Stefan Zimmerli, Thomas Jampen, Torsten Braun Architectural Issues of a Remote Network Laboratory. In Proceedings of Networked Learning Conference 2002, Berlin, Germany, May 2002

(Steinemann et Braun, 2002) Marc-Alain Steinemann et Torsten Braun. Remote versus Traditional Learning in a Computer Networks Laboratory. In Proceedings of IASTED

International Conference on Communications and Computer Networks, Cambridge, MA, USA, November 2002.

(Steinemann et al, 2002) Marc-Alain Steinemann, Stefan Zimmerli, Thomas Jampen, Torsten Braun Didactical Issues of a Remote Network Laboratory. In Proceedings of 4th International Conference on New Educational Environments, Lugano, Switzerland, May 2002.

(Tchounikine, 2002) Pierre Tchounikine. Pour une ingénierie des Environnements Informatiques pour l'Apprentissage Humain. In: Revue I3 Information Interaction Intelligence Vol. 2(1), p. 59-95.

(Tchounikine, 2009) Pierre Tchounikine, précis de recherche en ingénierie des EIAH, 2009 disponible en ligne sur <http://membres-liglab.imag.fr/tchounikine/Articles/PrecisV1.pdf> dernier accès: Mai 2010

(Tee et al, 2006) Kimberly Tee, Saul Greenberg, Carl Gutwin. Artifact awareness through screen sharing for distributed groups. Int. J. Human-Computer Studies 67 (2009) 677–702

(Tele-Lab IT-Security, 2010) site web du projet Tele-Lab IT-Security, disponible en ligne sur http://www.hpi.uni-potsdam.de/meinel/projekte/security_labs/tele_lab_it_security.html , dernier accès juin 2010

(Temperman et al, 2007) Gaëtan Temperman, Christian Depover, Bruno de Lièvre. Le tableau de bord, un outil d'awareness asynchrone. Actes de la conférence Environnements Informatiques Pour l'apprentissage Humain, Lausanne 2007, 359-370

(Van Joolingen et Zacharia, 2009) Wouter R. van Joolingen and Zacharias C. Zacharia Developments in Inquiry Learning. In N. Balacheff et al. (eds.), *Technology-Enhanced Learning*, 2009. P. 21-37. DOI 10.1007/978-1-4020-9827-7 2, C_ Springer Science+Business Media B.V. 2009

(VITELS, 2004) site web du projet VITELS disponible en ligne sur : <http://www.vitels.ch>, dernière mise à jours : 22 04 2004, dernier accès juin 2010

(VMWARE, 2007) VMWARE, Vmware Networking Concepts, disponible en ligne sur http://www.vmware.com/files/pdf/virtual_networking_concepts.pdf , dernière mise à jours : 2007, dernier accès : juillet 2010

(Vygotsky, 1985) Vygotski L. S., Pensées et langages 2ème édition, Éditions Sociales, 1985.

(Walter et al, 2007) Walter M. Fuertes, Jorge E. López de Vergara. A Quantitative Comparison of Virtual Network Environments Based on Performance Measurements. Poster at the 14th Workshop of the HP Software University Association, July 2007

(Weston et al, 1999) Cynthia Weston, Terry Gandell, Lynn McAlpine et Adam Finkelstein Designing Instruction for the Context of Online Learning, The Internet and Higher Education 2(1): 35- 44, ISSN: 1096-7516, Copyright D 1999 Elsevier Science Inc.

(White et al, 2002a) Brian White, Shashi Guruprasad, Mac Newbold, Jay Lepreau Leigh Stoller, Robert Ricci, Chad Barb, Mike Hibler, Abhijeet Joglekar. Netbed: An Integrated Experimental Environment. ACM SIGCOMM Computer Communication Review. Volume 32 , Issue 3 (July 2002) Pages: 27 – 27. ISSN:0146-4833

(White, et al, 2002b) Brian White, Jay Lepreau, Leigh Stoller, Robert Ricci, Shashi Guruprasad Mac Newbold, Mike Hibler, Chad Barb, Abhijeet Joglekar. An Integrated Experimental Environment for Distributed Systems and Networks. In 5th Symposium on Operating Systems Design & Implementation, pp. 255-270, December 2002

(Willems, 2008) Christian Willems Tele-Lab IT-Security: an Architecture for an online virtual IT Security Lab. In Proc. 2nd International Workshop on e-Learning and virtual and remote Laboratories (VLAB), Potsdam, Germany, 2008, ISBN: 978-3-940793-17-1

(Wulff et al, 2008) Markus Wulff, Patrick et Lauer Torsten Braun, Content management and architectural issues of a remote learning laboratory 2nd International Workshop on e-learning and Virtual and Remote Laboratories 2008, Hasso-Plattner-Institute Potsdam, Germany, February 14 - 15, 2008

(Yoo et al, 2006) Sun-Mi Yoo, James Won-Ki Hong, Jong-Geun Park , Chang-Won Ahn, Seong-Woon Kim, Performance Evaluation of WBEM Implementations, KNOM Review Vol. 8, No. 2, February 2006

(Zacharia et Olympiou, 2010) Zacharias C. Zacharia, Georgios Olympiou, Physical versus virtual manipulative experimentation in physics learning. *Learning and Instruction*, (2010) 1-15, doi:10.1016/j.learninstruc.2010.03.001

(Zimmerli et al , 2003) Stefan Zimmerli, Marc-Alain Steinemann, Torsten Braun. Resource Management Portal for Laboratories Using Real Devices on the Internet. SIGCOMM Computer Communication Review, Volume 33 , Issue 3 (July 2003) COLUMN: Educational environments Pages: 145 - 151: 2003 ISSN:0146-4833

Annexes

Annexe A : Synthèse de l'état de l'art des projets de télé-TPs existants

C	Critère	PEARL	Lab@Future	FORMID	ICCT@LAB	eINST	RoboTeach	VITELS	Tele-LAB
1.	1. Objectifs pédagogiques	1, 2, 3,4	1, 2, 3,4	1, 2,3	1, 2, 3,4	1, 2, 3,4	1, 2, 3,4	1, 2,3	1, 2,3
	2. Organisme de formation	4	1,2,3	1,2,3	4	4	1,2,3	4	4
	3. Publics visés	4	1,2,3	1,2,3	4	4	1,2,3	4	4
	4. Domaine d'enseignement	Générique	Générique	Générique	Générique	Générique	Robotique P.	Informatique	Informatique
2.	1. Type	1	4	2	1	1	1	1	3
	2. Architecture	Client/serveur	Client/serveur	Client/serveur	Trois-tiers	Trois-tiers	Client/serveur	Client/serveur	Client/serveur
	3. Middleware	CORBA	/	/	XML-RPC	JMS	/	/	/
	4. Interface et protocole d'instrumentation	Hors bande	Hors bande	/	Hors bande	Hors bande	Hors bande	Hors bande	In bande
	5. IHM de télé-opération	Client WEB/ adaptable	Client WEB/ adaptable	Client WEB	Client WEB	Client WEB/ adaptable	Client léger	Client WEB	Client WEB
	6. Modélisation du laboratoire	XML	XML	Texte	OWL	OWL	/	/	/
4.	1. Spécifique/Générique	Générique (WebCT)	Spécifique	Spécifique	Générique (Moodle)	Générique	Spécifique	Générique (WebCT)	Spécifique
	2. Interaction avec le laboratoire/ Interaction avec l'expérience	Via le modèle XML	Contrôle direct	Via le moniteur IMS-LD	Via le moniteur IMS-LD	Contrôle direct	Contrôle directe	Contrôle directe	Contrôle directe
	3. Interaction avec le télé-TP	Moniteur LabScript	Cours Lab@Future	Moniteur IMS-LD	Moniteur IMS-LD	/	Moniteur du projet	Cours WebCT	Tuteur intelligent/

				intégré	tiers				Hypermédia
--	--	--	--	---------	-------	--	--	--	------------

C	Critère	PEARL	Lab@Future	FORMID	ICCT@LAB	eINST	RoboTeach	VITELS	Tele-LAB
5.	1. Théorie(s) d'apprentissage(s) sous-jacentes	Socioconstructivisme	Constructivisme / Apprentissage expansif	Constructivisme	Neutre	Socioconstructivisme	Socioconstructivisme	Constructivisme	Constructivisme
	2. Stratégie et pratiques pédagogiques	/	/	Exercices	/	/	Projets	Exercices, Résolution de problème	Exercices, Résolution de problème
	3. Scénarisation pédagogique	Opérationnel en XML	/	opérationnel en IMS-LD	opérationnel en IMS-LD	/	Opérationnelle	Informelle	Informelle : style magistral
	4. Outils d'aide à la conception pédagogique	Guides de rédaction, feuilles de style	L'outil AODM (Théorie de l'activité)	L'outil auteur OASIS	L'éditeur IMS-LD Reload	/	Editeur de projets de SPLACH	Guides de rédaction, feuilles de style	/
6.	1. Outils de communication et de collaboration	Outils de communications/collaboration basiques (WebCT)	Environnement sociale de réalité mixte/réalité virtuelle : les	Outils de communications/collaboration basiques	Outils de communications et collaboration basiques (Moodle)	Messagerie instantanée	Outils de communication/collaboration, rédaction collaborative de documents	Les outils offerts par WebCT	/

	2. Support de l'Awareness	Indicateur de présence	acteurs sont présents virtuellement dans le laboratoire	/	Outils Moodle	Indicateur de présence, Gestion des accès concurrents au laboratoire	Partage de l'environnement d'expérimentation	/	/
--	---------------------------	------------------------	---	---	---------------	--	--	---	---

C	Critère	PEARL	Lab@Future	FORMID	ICCT@LAB	eINST	RoboTeach	VITELS	Tele-LAB
7.	1. Outils de perception des activités individuelles et/ou collectives des apprenants	Consultation des traces des opérations sur les équipements, Consultation des traces du LabScript (l'activité)	Environnement sociale de réalité mixte/réalité virtuelle : les acteurs sont présents virtuellement dans le laboratoire, ils peuvent communiquer librement et manipuler directement des	Suivi en temps réel de l'état global et détaillé de la progression des apprenants en classe virtuelle	Outils de communication et de collaboration de Moodle	- Outils de communication et de collaboration - Consultation journal des actions sur les équipements distants	Consultation de la progression des apprenants sur tout le parcours et sur chaque activité Consultation des productions	les outils de com. WebCT	/
	2. Outils de suivi synchrone individuel et/ou collectif des apprenants	Outils de communication de WebCT, accès en contrôle sur le laboratoire		Outils de communication et collaboration		/	Partage de l'environnement d'apprentissage Partage du dispositif	FAQ, Forum, Messagerie, Téléphone	/

	3. Outils de gestion du suivi des apprenants	Gestion de la queue des questions, Diffusion de messages aux apprenants	objets virtuels connectés à des dispositifs réels	Gestion des appels des apprenants		/	Annotation des interventions audiovisuelles Gestion des appels des apprenants		/
--	--	---	---	-----------------------------------	--	---	--	--	---

Annexe B : Services WEB dédiés à la gestion de laboratoires en ligne

Identifiant du service	Paramètres d'entrée	Opérations WBEM, objets CIM manipulés et résultats retournés
RegisterLab	<ul style="list-style-type: none"> - Tableau des paramètres d'identification et de localisation d'un laboratoire - Tableau des types de ressources mises à dispositions, pour chaque type le nombre de ressources - Le paramètre booléen Remote, lorsqu'il est fixé à <i>True</i> indique qu'il s'agit d'un laboratoire partagé par une autre institution - Le paramètre <i>WBEMName</i> indique le nom ou l'adresse IP du serveur WBEM qui gère le laboratoire (lorsque le paramètre <i>Remote</i> est fixé à <i>True</i>) 	<ul style="list-style-type: none"> - Création d'une instance <i>LAB_Domain</i> et initialisation de ses attributs avec les paramètres d'entrée (voir tableau 7-1) - Création des instances CIM des ressources (Tableau 7-2) - Création des instances <i>LAB_ResourcePool</i> (tableau 7-3) - Création de l'association <i>HostedResourcePool</i> entre <i>LAB_ResourcePool</i> et <i>LAB_Domain</i> - Création des associations <i>Component</i> entre <i>LAB_ResourcePool</i> et les instances de ressources - Lorsqu'il s'agit d'un laboratoire partagé par une autre institution, interroger le serveur WBEM dont le nom a été fourni en paramètre pour obtenir toutes les classes précédentes - Retourne <i>True</i> en cas de succès ou <i>False</i> autrement
ModifyLabStatus	Même paramètres que la fonction précédente, sauf que l'identifiant du laboratoire doit correspondre à un laboratoire préalablement enregistré.	<ul style="list-style-type: none"> - Mise à jour des instances des classes existantes - Créations des instances non existantes avec les associations correspondantes, par exemple de nouveaux types de ressources (<i>LAB_ResourcePool</i>) et de nouvelles ressources. - Retourne <i>True</i> en cas de succès ou <i>False</i> autrement
DeleteLab	Identifiant du laboratoire	<ul style="list-style-type: none"> - Suppression de tous les objets CIM des classes : <i>LAB_Domain</i>, <i>LAB_ResourcePool</i>, et les instances de ressources ainsi que les différentes associations qui les relient

		- Retourne <i>True</i> en cas de succès ou <i>False</i> autrement
GetLabsList	Aucun	- Retourne la liste des identifiants des laboratoires enregistrés éventuellement, avec les noms des serveurs WBEM locaux qui les gèrent.
GetLabStatus	Identifiant du laboratoire	- Retourne l'état d'allocation des ressources d'un laboratoire sous la forme d'un tableau qui contient pour chaque Type de ressource, le nombre d'instances utilisés et le nombre d'instances disponibles.
AllocateLabResource	Identifiant du laboratoire, identifiant du type de ressource et nombre d'instances à réserver, éventuellement la liste des identifiants des ressources à réserver	<ul style="list-style-type: none"> - Marquer la ressource comme réservée en modifiant l'attribut correspondant dans son instance CIM (voir tableau 7-2) - Mettre à jours l'instance <i>LAB_ResourcePool</i> en mettant à jours l'attribut qui indique nombre d'instances disponibles et en augmentant l'attribut qui indique le nombre d'instances occupées (voir tableau 7-3) - Retourne la liste des identifiants des ressources réservées
FreeLabResource	Mêmes paramètres que la fonction précédente	<ul style="list-style-type: none"> - Mettre à jours les instances <i>LAB_ResourcePool</i> et ceux des ressources elle-même pour indiquer qu'elles sont libre (voir tableau 7-2 et 7-3) - Retourne <i>True</i> en cas de succès ou <i>False</i> autrement

Annexe C : Services WEB dédiés à la gestion des expériences en ligne

Identifiant du service	Paramètres d'entrée	Opérations WBEM, objets CIM manipulés et résultats retournés
CreateExperiment	<ul style="list-style-type: none"> - Identifiant de l'expérience - Le modèle d'initialisation de l'expérience (en format XML) qui contient les valeurs de tout ou partie des attributs. 	<ul style="list-style-type: none"> - Création de l'instance <i>LAB_Experiment</i> correspondante à l'expérience à construire. - Invocation de la méthode <i>Create ()</i> de l'objet créé qui effectue les tâches suivantes : <ul style="list-style-type: none"> - Exécution de l'algorithme d'identification des ressources nécessaires à partir du modèle de l'expérience - Exécution de l'algorithme de recherche des ressources disponibles pour construire l'expérience - Invocation des services de réservation de ressources nécessaires à l'expérience (voir le tableau 6-4) - Etablissement des liens <i>Component</i> et/ou <i>SystemComponent</i> entre l'instance <i>LAB_Experiment</i> et les ressources allouées - Eventuellement, créer les instances de classes et associations supplémentaires spécifiques à la gestion de l'expérience - Retourne le même modèle de l'expérience fourni comme paramètre d'entrée avec les valeurs des identifiants et les autres attributs fixés à l'initialisation en cas de succès, sinon rend un message d'erreur.
StartExperiment	Identifiant de l'expérience	<ul style="list-style-type: none"> - Invocation de la méthode <i>Start ()</i> de l'objet CIM <i>LAB_Experiment</i> qui exécute les méthodes d'initialisation des ressources réservées si cela n'a pas été fait par la méthode de création et donne la main pour l'exploitation de l'expérience. - Retourne <i>True</i> en cas de succès et <i>False</i> sinon
ExcuteCommand	<ul style="list-style-type: none"> - Identifiant de l'expérience - Nom de la commande et la liste de 	<ul style="list-style-type: none"> - Exécution d'une commande sur une ou plusieurs ressources de l'expérience - Retourne <i>True</i> en cas d'exécution et <i>False</i> sinon

	ses arguments et options	
ReceiveMessage	Identifiant de l'expérience	Réception de la réponse à une commande ou un événement généré par l'expérience.
SuspendExperiment	Identifiant de l'expérience	Invocation de la méthode <i>Suspend()</i> de l'objet CIM de l'expérience qui arrête l'expérience en question et en libère toutes les ressources
SaveExperiment	Identifiant de l'expérience	<ul style="list-style-type: none"> - Invocation de la méthode <i>Save()</i> de l'objet CIM de l'expérience qui en sauvegarde l'état de toutes les instances et association dans le Référentiel. - Retourne l'identifiant de la sauvegarde qui sera utilisée par la méthode de restauration, en cas d'échec retourne 0.
ResumeExperiment	Identifiant de l'expérience	<ul style="list-style-type: none"> - Invocation de la méthode <i>Resume()</i> de l'objet CIM de l'expérience qui va redémarrer l'expérience suspendue dans l'état où elle était lors de sa suspension. - Retourne <i>True</i> en cas de succès et <i>False</i> sinon
RestoreExperiment	<ul style="list-style-type: none"> - Identifiant de l'expérience - Identifiant de la sauvegarde de l'expérience 	<ul style="list-style-type: none"> - Invocation de la méthode <i>Restore()</i> de l'objet CIM de l'expérience qui va redémarrer l'expérience dans l'état correspondant à l'identifiant de la sauvegarde fourni - Retourne <i>True</i> en cas de succès et <i>False</i> sinon
DeleteExperiment	Identifiant de l'expérience	<ul style="list-style-type: none"> - Invocation de la méthode <i>Delete()</i> de l'objet CIM de l'expérience qui exécute toutes les opérations nécessaires à la suppression de l'expérience en commençant par l'arrêter, en libérer les ressources et supprimer toutes les instances CIM relatives - Retourne <i>True</i> en cas de succès et <i>False</i> sinon
GetExperimentsList	Aucun	Retourne la liste de toutes les expériences créées (qu'elles soient en arrêt ou en marche)
GetExperimentStatus	<ul style="list-style-type: none"> - Identifiant de l'expérience - Identifiant d'une ressource 	<ul style="list-style-type: none"> - Retourne l'état complet de l'expérience à travers son modèle qui contient les valeurs actuelle de tous les attributs des objets CIM à un moment donnée

	(optionel)	- Lorsque l'identifiant d'une ressource est fourni, retourne uniquement l'état de l'objet CIM correspondant.
--	------------	--

Annexe D : Conception, construction et déploiement de machines virtuelles

D.1 Phase de conception

La conception d'une machine virtuelle consiste en deux opérations : renseigner ses caractéristiques matérielles et logicielles dans un modèle descriptif et construire son système de fichiers. Le Tableau D-1 donne des exemples des caractéristiques les plus communes.

Tableau D-1: Paramètres de configuration d'une machine virtuelle

Caractéristiques	Valeurs	Options
Architecture du processeur	I386, I686, X86_64 etc.	Nombre de processeurs
Taille de la RAM	128M	
Type d'installation	Fichier image, Répertoire LVM	
Espace disque de la partition principale	Point de montage	Taille
Espace disque du SWAP	Taille (512 M)	
Partitions supplémentaires	Point de Montage	Taille
Interface(s) réseau	Type et nombre de l'interface : Ethernet, Wifi autres.	
Autres périphériques	Carte son, carte graphique,	
Système de fichier	NTFS, VFAT, EXT2/3/4, ReiserFS	
Système d'exploitation	Windows, Unix, Linux (version du noyau)	XP, Vista, Server 2003, Solaris,, BSD, Ubuntu, Redhat, Debian ..
Identification de la machine	Nom, Nom de domaine	
Paramètres IP par interface réseau	Adresse dynamique, Adresse statique	Adresse IP, masque de sous réseau, adresse passerelle, adresse du serveur de nom
Logiciels supplémentaires	MS Office, Open Office, Eclipse	

Chaque technologie de virtualisation possède son propre langage de modélisation, le plus souvent fondé sur une représentation textuelle. Mais dû aux enjeux économiques importants liés à la virtualisation, le DMTF vient de publier un standard de modélisation des machines virtuelles, fondé sur le langage XML. Ce méta-modèle appelé *Open Virtual Format* ou OVF⁸⁹, permet de spécifier les caractéristiques les plus fines d'une machine virtuelle. Adopté par la plupart des acteurs (VMWARE, Microsoft, Oracle, CITRIX, Open source etc.) il permet de faciliter la gestion du cycle de vie des machines virtuelles et leurs migrations entre les différentes technologies.

Pour faciliter la conception des machines virtuelles, des outils auteurs fondés sur le standard OVF sont déjà mis à disposition comme OpenOVF d'IBM⁹⁰, Kensho de CITRIX⁹¹, VMWARE OVF Tools etc. Mais il est également possible d'utiliser un éditeur XML standard (ou en développer un au besoin) fondé sur le DTD-XML de ce langage. Mais dû à la complexité de ce langage, souvent la phase de conception est rattachée à la phase de construction à travers des outils spécialisés qui offrent une interface utilisateur intuitive.

D.2 Phase de construction

Cette phase consiste en la construction du système de fichiers de la machine virtuelle avec ses différentes partitions. Durant cette étape, le fichier de configuration est vérifié syntaxiquement et sémantiquement en s'assurant également que le système dispose d'un espace disque suffisant pour cette opération.

Cette phase peut servir à créer des machines virtuelles modèles pour les distribuer et en faire autant de copies que de machines virtuelles. Avant cela il est également possible de démarrer la machine virtuelle de référence, lui apporter des modifications, puis l'arrêter, la sauvegarder et la distribuer.

D'un point de vue pédagogique ceci apporte beaucoup de bénéfices :

⁸⁹ <http://www.vmware.com/appliances/getting-started/learn/ovf.html> ,

<http://www.dmtf.org/standards/mgmt/vman/>

⁹⁰ <http://open-ovf.sourceforge.net>

⁹¹ <http://community.citrix.com/display/xs/Kensho>

- Un enseignant va se concentrer à préparer une seule machine virtuelle type pour une télé-TP donné, puis il va la dupliquer pour tous les apprenants.
- Les apprenants peuvent commettre des erreurs sans « avoir peur » des conséquences, mettant en œuvre ainsi le principe de l'apprentissage par essais/erreurs, car ils savent qu'ils peuvent revenir en arrière à chaque fois qu'ils le souhaitent, juste par la réinitialisation de l'expérience au lieu de la fastidieuse opération de réinstallation des machines physiques.
- Du point de vue de la maintenance et de la sécurité, les éventuels risques liés à l'utilisation des machines virtuelles sont éliminés dès que les clones sont supprimés.

Pour assister le concepteur dans cette phase, un grand nombre d'outils existent, nécessitant souvent un serveur dédié qui va servir également comme dépôt aux images de machines virtuelles. A titre d'exemple citons les outils propriétaires VMWARE, CITRIX et MS HyperV et les outils génériques comme VIRT-MANAGER⁹², VMBUILDER⁹³, ENOMALISME⁹⁴ et RBUILDER⁹⁵ qui offrent une interface graphique intuitive pour la modélisation des machines virtuelles et génèrent au passage les modèles OFV associés d'une façon transparente à l'utilisateur.

Notons enfin que des sites spécialisés comme <http://www.easyvmx.com/>, <http://stacklet.com/>, <http://www.turnkeylinux.org/>, <http://www.vmware.com/appliances/directory/> et <http://wiki.rpath.com/wiki/Appliance> mettent à disposition des images de systèmes invités pour différentes technologies et d'autres sites comme <http://www.rpath.com> permettent de les construire en ligne puis les télécharger.

Parmi ces outils, nous avons sélectionné *Rbuilder* car c'est le seul outil open source et gratuit avec une interface WEB sophistiquée (donc intégrable dans notre LMS Moodle) qui permet la gestion de tout le cycle de vie des machines virtuelles (sauf le déploiement laissé à la couche Expérimentation). L'autre avantage de cet outil est sa compatibilité avec plusieurs technologies de virtualisation et sa capacité à personnaliser les applications à installer sur les

⁹² <http://virt-manager.et.redhat.com/>

⁹³ <http://dcgrendel.thewaffleiron.net/vmbuilder/>

⁹⁴ <http://www.enomaly.com/>

⁹⁵ <http://www.rpath.com/>

machines virtuelles et de gérer plusieurs versions de la même machine. Il permet également de créer des images de disques d'installation en format ISO afin de déployer des expériences sur des équipements physiques.

D.3 Phase de déploiement

Une fois une machine virtuelle est construite via *Rbuilder*, elle est disponible en téléchargement via une URL web fourni par cet outil. Pour la déployer pour la première fois il faudrait indiquer à *Rbuilder* l'adresse de notre serveur d'expérimentation ou bien copier manuellement l'image dans ce serveur et lancer son exécution. Durant cette opération le système s'assure que les ressources demandées sont disponibles avant de les allouer à la nouvelle machine virtuelle.

D.4 Phase d'exploitation

Après le démarrage de l'expérience, l'utilisateur peut y accéder pour l'exploiter à travers l'interface dédiée mis à sa disposition au niveau de la couche Apprentissage

D.5 Phases d'arrêt et de suspension

Lorsque l'utilisateur n'a plus besoin de la machine ou lorsque sa session a expirée, il peut l'arrêter lui-même ou c'est le système qui le fait automatiquement pour en libérer les ressources qu'elle occupe. Dans un contexte pédagogique, le démarrage et l'arrêt d'une expérience est soumis à la planification des séances de télé-TP.

D.6 Phase de sauvegarde et d'archivage

Une machine virtuelle peut être sauvegardée et archivée à un moment donnée lorsqu'on estime qu'elle a atteint un état significatif et ce pour différents objectifs:

- Un apprenant étant arrivé à une situation stable ou satisfaisante dans son télé-TP, peut demander l'archivage de sa machine avant de continuer son activité. Lorsqu'il rencontre une situation de blocage, il peut demander la restauration de sa machine sauvegardée dans un état donné au lieu de reprendre du début.
- A la fin d'un télé-TP, l'enseignant sauvegarde la machine virtuelle de chaque apprenant ou groupe d'apprenants pour effectuer l'évaluation de leurs travaux pratiques.

Dans notre contexte, ceci peut être réalisé grâce à la fonctionnalité de gestion de versions des machines virtuelles de *Rbuilder*.

D.7 Phase de suppression

La suppression d'une machine virtuelle, consiste à en supprimer le système de fichier situé dans le vivier des machines virtuelles de *Rbuilder*.

Annexe E : Diagramme de l'expérience type en informatique selon quatre formats

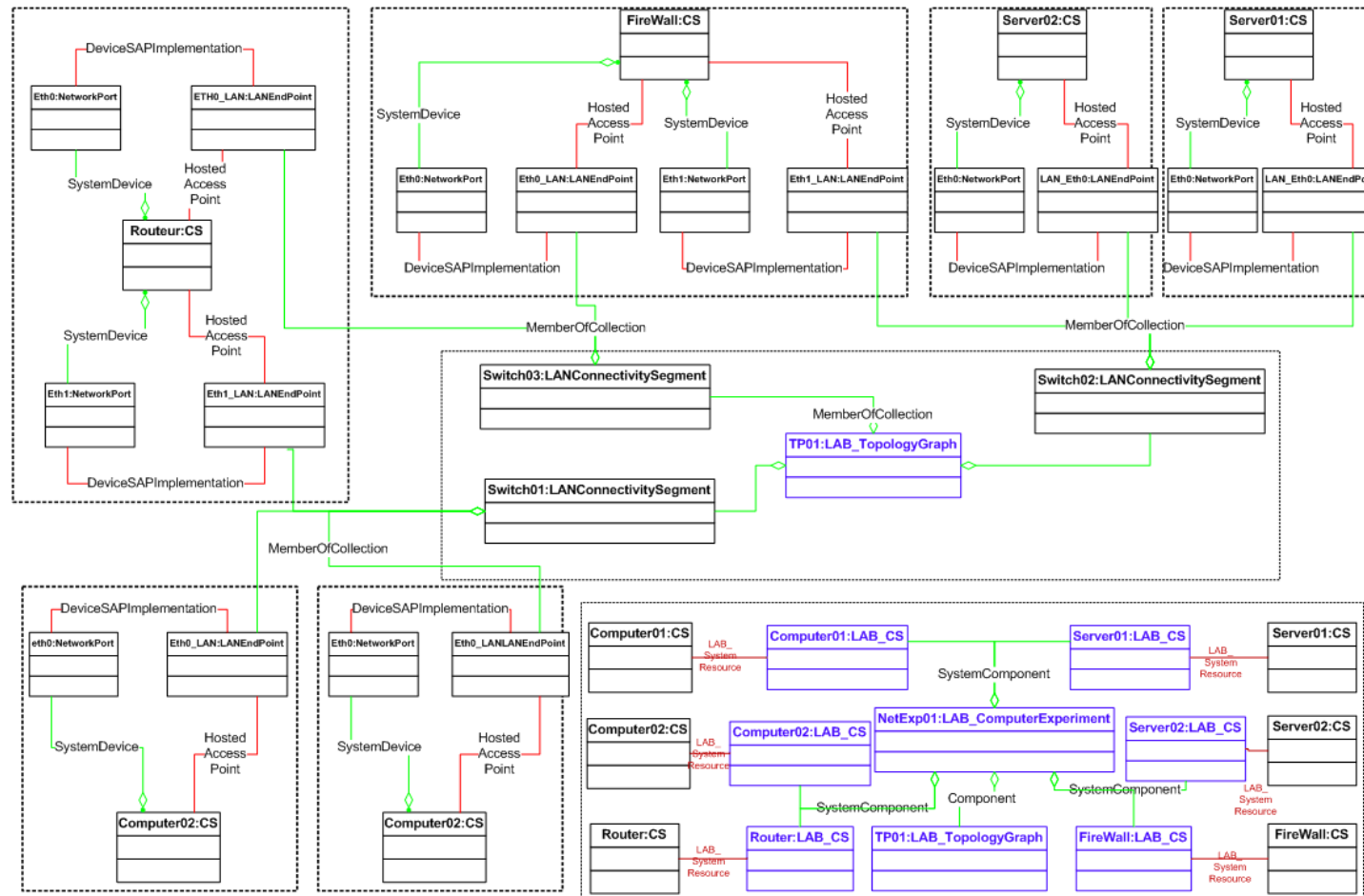


Figure E-1 : Diagramme UML des instances CIM du modèle de l'expérience (en bas à gauche le modèle globale)

Tableau E-1: Quelques instances CIM en format MOF de l'expérience

```

instance of TEL_ComputerExperiment {
  ElementName = "NetExp01";
  Identifier = " NetExp01";
};
instance of TEL_SystemComponent {
  GroupComponent = "TEL_ComputerExperiment.Identifier=\\" NetExp01\\"";
  PartComponent                                     =
"TEL_ComputerSystem.CreationClassName=\\"TEL_ComputerSystem\\",Name=\\"Computer01.NetExp01\\"";
};
instance of TEL_SystemComponent {
  GroupComponent = "TEL_ComputerExperiment.Identifier=\\"NetExp01\\"";
  PartComponent                                     =
"TEL_ComputerSystem.CreationClassName=\\"TEL_ComputerSystem\\",Name=\\"Computer02.NetExp01\\"";
};
instance of TEL_SystemComponent {
  GroupComponent = "TEL_ComputerExperiment.Identifier=\\"NetExp01\\"";
  PartComponent                                     =
"TEL_ComputerSystem.CreationClassName=\\"TEL_ComputerSystem\\",Name=\\"gateway.NetExp01\\"";
};
instance of TEL_SystemComponent {
  GroupComponent = "TEL_ComputerExperiment.Identifier=\\"NetExp01\\"";
  PartComponent                                     =
"TEL_ComputerSystem.CreationClassName=\\"TEL_ComputerSystem\\",Name=\\"Server1.NetExp01\\"";
};
instance of TEL_SystemComponent {
  GroupComponent = "TEL_ComputerExperiment.Identifier=\\"NetExp01\\"";
  PartComponent                                     =
"TEL_ComputerSystem.CreationClassName=\\"TEL_ComputerSystem\\",Name=\\"Server2.NetExp01\\"";
};
instance of TEL_SystemComponent {
  GroupComponent = "TEL_ComputerExperiment.Identifier=\\"NetExp01\\"";
  PartComponent                                     =
"TEL_ComputerSystem.CreationClassName=\\"TEL_ComputerSystem\\",Name=\\"Firewall.NetExp01\\"";
};
instance of TEL_Component
{

```

```

GroupComponent = "MLN_Project.Name=\"NetExp01\"";
PartComponent = ":MLN_TopologyGraph.InstanceID=\"TopologyGraph_NetExp01\"";
};
instance of TEL_TopologyGraph {
ElementName = "TopologyGraph_expv03"; InstanceID = "TopologyGraph_expv03";};
instance of TEL_ComputerSystem {
ElementName = "Computer01"; CreationClassName = "TEL_ComputerSystem"; Name = "Computer01"; };
instance of TEL_ComputerSystem {
ElementName = "Computer02"; CreationClassName = "TEL_ComputerSystem"; Name = " Computer02";};
instance of TEL_ComputerSystem{
ElementName = "Server01"; CreationClassName = "TEL_ComputerSystem";Name = " Server01";};
instance of TEL_ComputerSystem{
ElementName = "Server02"; CreationClassName = "TEL_ComputerSystem"; Name = " Server02";};
instance of TEL_ComputerSystem{
ElementName = "Router";CreationClassName = "TEL_ComputerSystem";Name = " Router ";};
instance of TEL_ComputerSystem{
ElementName = "Firewall";CreationClassName = "TEL_ComputerSystem";Name = " Firewall ";};

```

Tableau E-2 : Modèle de l'expérience en format MLN

```
#include /opt/mln/projects/root/superclass.mln
global {
    project NetExp01
}

host Server01 {
    superclass experimentcommon
    template ubuntu.9-04.x86.ext3
    network eth0 {
        bridge mgmt
        address dhcp
        mac 00:50:56:1d:b8:c1
    }
    network eth1 {
        mac 00:50:56:24:d4:dc
        switch Switch01
    }
}

host Server02 {
    superclass experimentcommon
    template ubuntu.9-04.x86.ext3
    network eth0 {
        bridge mgmt
        address dhcp
        mac 00:50:56:1c:4f:dd
    }
    network eth1 {
        mac 00:50:56:2a:22:62
        switch Switch01
    }
}

switch Switch01 {
    xen 1
    bridge swA-expe-v01
}

switch Switch02 {
    xen 1
    bridge swB-expe-v01
}

switch Switch02 {
    xen 1
    bridge swC-expe-v01
}

host Router {
    superclass experimentcommon
    template ubuntu.9-04.x86.ext3
    network eth0 {
        bridge mgmt
        address dhcp
        mac 00:50:56:3f:17:06
    }
    network eth1 {
        mac 00:50:56:28:43:ae
        switch Switch02
    }
    network eth2 {
        mac 00:50:56:1c:28:75
        switch Switch03
    }
}

host Computer01 {
    superclass experimentcommon
    template ubuntu.9-04.x86.ext3
    network eth0 {
        bridge mgmt
        address dhcp
    }
}
```



```

    }
}
host Firewall {
    superclass experimentcommon
    template ubuntu.9-04.x86.ext3
    network eth0 {
        bridge mgmt
        address dhcp
        mac 00:50:56:2e:21:83
    }
    network eth1 {
        mac 00:50:56:03:fb:ce
        switch Switch01
    }
    network eth2 {
        mac 00:50:56:13:b5:e1
        switch Switch02
    }
}
}

```

```

        mac 00:50:56:18:af:98
    }
    network eth1 {
        mac 00:50:56:2b:d8:66
        switch Switch03
    }
}
host Computer02 {
    superclass experimentcommon
    template ubuntu.9-04.x86.ext3
    network eth0 {
        bridge mgmt
        address dhcp
        mac 00:50:56:32:24:4d
    }
    network eth1 {
        mac 00:50:56:1a:fd:29
        switch Switch03
    }
}
}

```

Tableau E-3 : Modèle de l'expérience format Network WeatherMap

# Automatically generated by php-weathermap vPlus 0.4	# TEMPLATE-only NODEs:
KEYPOS DEFAULT -1 -1 Traffic Load	NODE NODE DEFAULT
KEYTEXTCOLOR 0 0 0	MAXVALUE 100
KEYOUTLINECOLOR 0 0 0	NODE SWITCH DEFAULTSWITCH
KEYBGCOLOR 255 255 255	MAXVALUE 100
BGCOLOR 255 255 255	NODE DESKTOP DEFAULTDESKTOP
TITLECOLOR 0 0 0	SET ph-settings 1:256M:0M
TIMECOLOR 0 0 0	SET os Linux:Ubuntu:9.10
SCALE DEFAULT 0 0 192 192 192	MAXVALUE 100
SCALE DEFAULT 0 1 255 255 255	NODE SERVER DEFAULTSERVER
SCALE DEFAULT 1 10 140 0 255	SET ph-settings 1:256M:0M
SCALE DEFAULT 10 25 32 32 255	SET os Linux:Ubuntu:9.10
SCALE DEFAULT 25 40 0 192 255	MAXVALUE 100
SCALE DEFAULT 40 55 0 240 0	NODE ROUTER DEFAULTROUTER
SCALE DEFAULT 55 70 240 240 0	SET os Linux-for-router:Debian-router:Debian+Quagga
SCALE DEFAULT 70 85 255 192 0	MAXVALUE 100
SCALE DEFAULT 85 100 255 0 0	NODE FIREWALL DEFAULTFIREWALL
# End of global section	SET os Linux-for-firewall:Debian-firewall:Debian+iptables
	MAXVALUE 100
	# TEMPLATE-only LINKs:
	LINK DEFAULT
	BANDWIDTH 100M
Description des nœuds	Description des liens
NODE node04416	LINK node04416-node04477
TEMPLATE SERVER	NODES node04416 node04477
LABEL Server01	LINK node04477-node04452
SET fs-settings /:ext3fs:4G:256M	NODES node04477 node04452
SET ph-settings 1:256M:4256M	LINK node04477-node04688
SET os Linux:Ubuntu:9.10	NODES node04477 node04688
SET network:ethernet:1	LINK node04688-node04658
SET ip-settings eth0:static: : : :	NODES node04688 node04658
POSITION 45 140	LINK node04658-node04715
NODE node04452	NODES node04658 node04715
TEMPLATE SERVER	LINK node04715-node04515

LABEL Server02	NODES node04715 node04515
SET fs-settings /:ext3fs:4G:256M	LINK node04546-node04515
SET ph-settings 1:256M:4256M	NODES node04546 node04515
SET os Linux:Ubuntu:9.10	LINK node04515-node04608
SET network:ethernet:1	NODES node04515 node04608
SET ip-settings eth0:static: : : :	
POSITION 45 425	
NODE node04477	
TEMPLATE SWITCH	
LABEL Switch01	
POSITION 45 275	
NODE node04515	
TEMPLATE SWITCH	
LABEL Switch03	
POSITION 750 275	
NODE node04546	
TEMPLATE DESKTOP	
LABEL PC01	
SET fs-settings /:ext3fs:4G:256M	
SET ph-settings 1:256M:4256M	
SET os Linux:Ubuntu:9.10	
SET network:ethernet:1	
SET ip-settings eth0:static: : : :	
POSITION 750 140	
NODE node04608	
TEMPLATE DESKTOP	
SET fs-settings /:ext3fs:4G:256M	
SET ph-settings 1:256M:4256M	
SET os Linux:Ubuntu:9.10	
SET network:ethernet:1	
SET ip-settings eth0:static: : : :	
LABEL PC02	
NODE node04658	
TEMPLATE SWITCH	
LABEL Switch02	
POSITION 400 275	
NODE node04688	

TEMPLATE FIREWALL

LABEL FireWall

SET fs-settings /:ext3fs:4G:256M

SET ph-settings 1:256M:4256M

SET os Linux:Ubuntu:9.10

SET network:ethernet:2

SET ip-settings eth0:static: : : :

SET ip-settings eth1:static: : : :

POSITION 220 275

NODE node04715

TEMPLATE ROUTER

LABEL Router

SET fs-settings /:ext3fs:4G:256M

SET ph-settings 1:256M:4256M

SET os Linux:Ubuntu:9.10

SET network:ethernet:2

SET ip-settings eth0:static: : : :

SET ip-settings eth1:static: : : :

POSITION 570 275

Annexe F : Mode d'emploi de l'interface de Télé-expérience

1. Introduction et mise en garde	348
2. Premier accès et vue d'ensemble	348
3. La messagerie instantanée	351
a. Le canal de l'équipe	352
b. Le canal du tuteur	352
c. Emoticons et actes de langages	352
4. La fenêtre d'expérimentation	352
a. Explorateur de l'expérience	353
b. Explorateur des utilisateurs	355
5. Une Activité type de Travaux Pratiques en Systèmes et Réseaux : Adressage et Configuration IP de base sous Linux	353

Introduction et mise en garde

L'objectif de cette application est de faciliter le télé-contrôle individuel et collaboratif d'expériences en informatique, en particulier les expériences en réseau caractérisées par une architecture plus ou moins complexe et composées de différents types de nœuds (machines, routeurs, *switchs*, etc.). Il s'agit donc, à travers une interface fondée sur les technologies du WEB 2.0 ne nécessitant qu'un navigateur WEB, de faire des travaux pratiques à distance.

Il est important de signaler que cette application n'a été testée qu'avec le navigateur **Mozilla Firefox**, et qu'il s'agit d'un prototype issu d'un travail de recherche et non d'un produit commercial ; nous vous demandons donc d'être indulgents si vous rencontrez des bugs, et de nous les signaler.

1. Premier accès et vue d'ensemble

L'accès à l'application s'effectue via la plate-forme Moodle (**Télé-TP en Systèmes et Réseaux Informatiques**) dans le cours « **Accès à l'Expérience en Ligne** » ; une description sommaire de l'activité figure dans cet espace de cours.

L'application est accessible en cliquant sur le lien « **Interface d'Accès à l'Expérience** » qui ouvre une nouvelle fenêtre illustrée par la Figure 1 ; nous vous conseillons d'afficher la fenêtre en mode plein écran afin de mieux distinguer le grand nombre d'éléments contenus dans l'IHM. Vous devez ensuite vous authentifier avec les identifiants fournis, aussi dans le cadre de gauche que dans celui de droite.

Utilisateur :
Mot de passe :

AJAX Chat

Nom d'utilisateur:
Mot de passe*:
Salon:
Langue:

* Utilisateurs enregistrés

AJAX
Chat
©
blueimp.net

Figure 1. Fenêtre d'accès à l'interface d'expérimentation

Une fois connecté, l'interface ressemblera à celle de la Figure 2 ci-dessous. Le cadre de droite propose une application de messagerie instantanée (*chat*), alors que le reste de la fenêtre contient les fonctionnalités propres à la réalisation du TP.

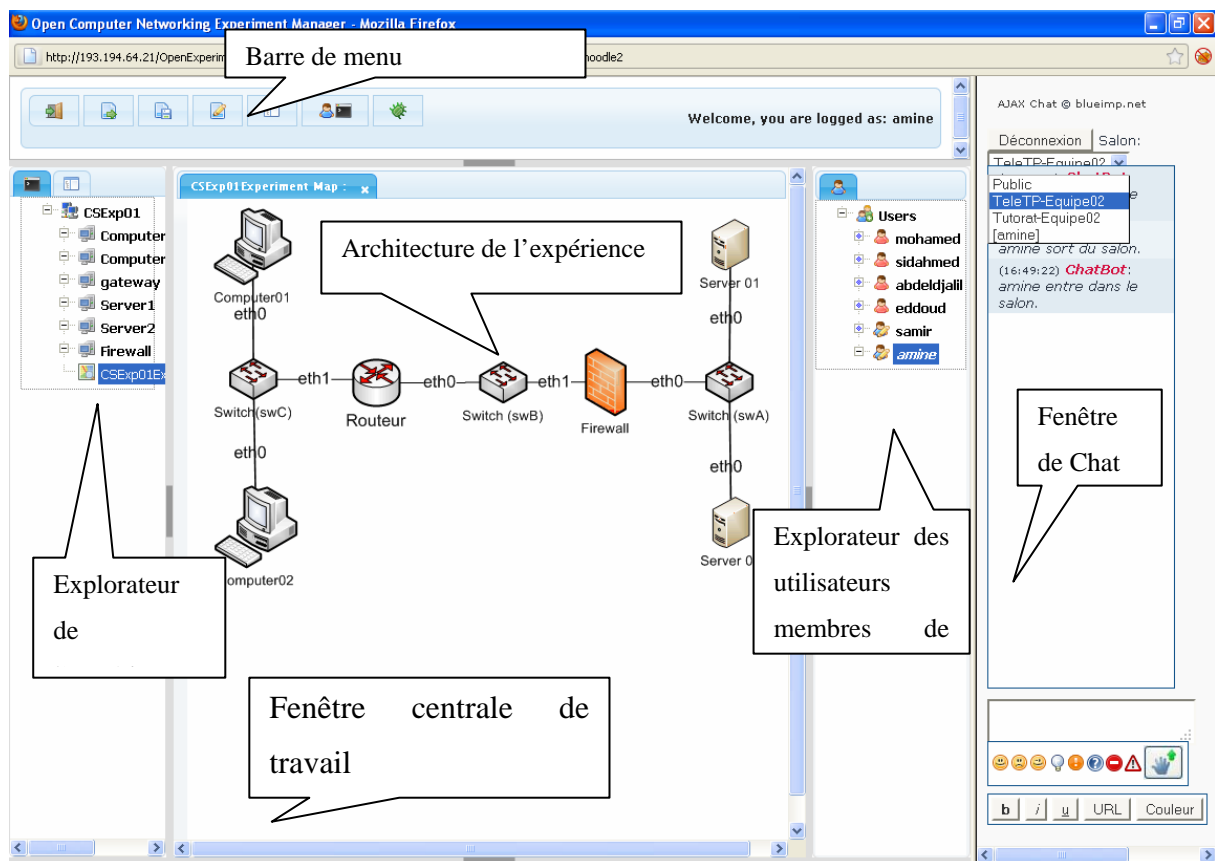


Figure 2. L'interface d'expérimentation après l'authentification

2. La messagerie instantanée

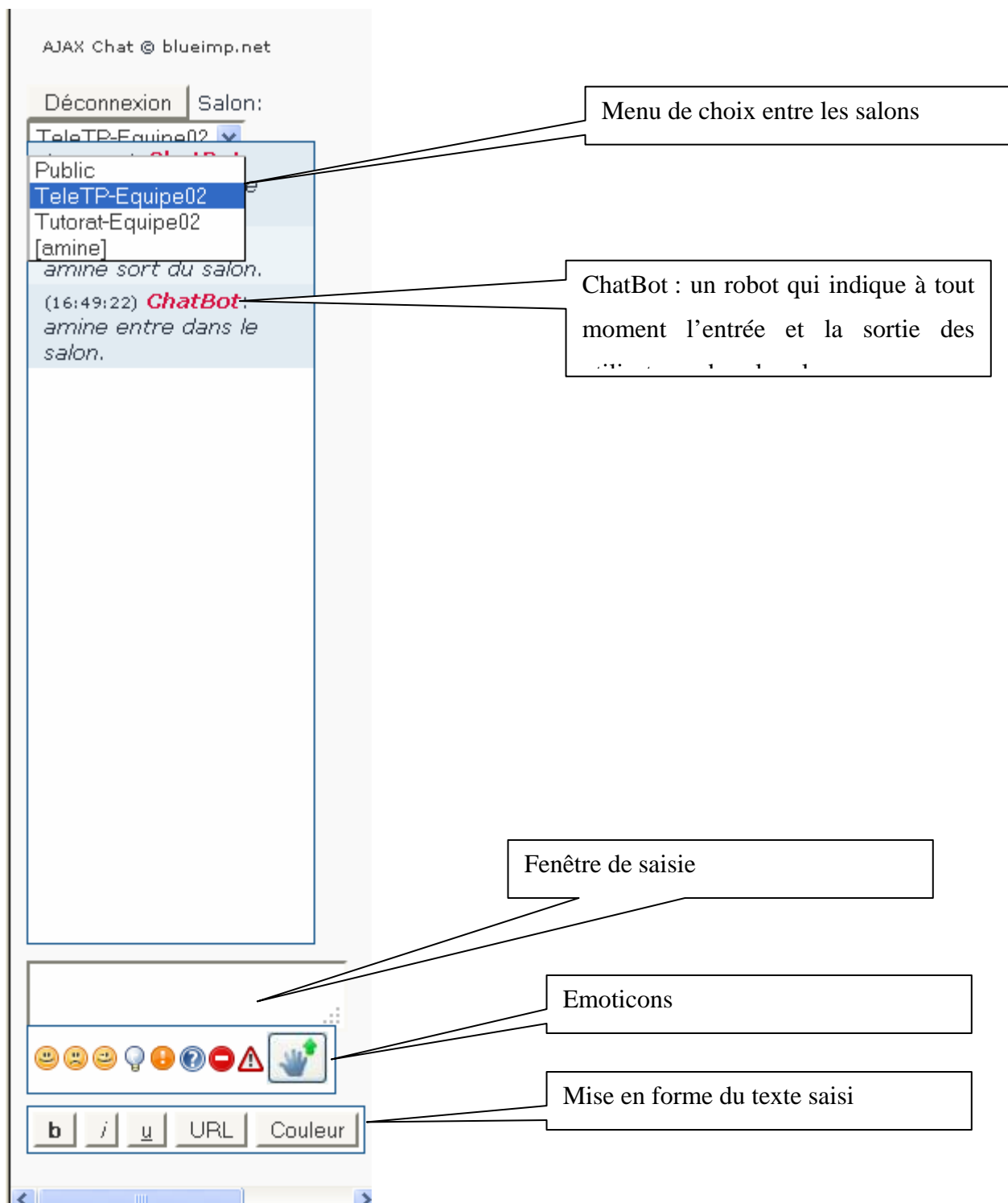


Figure 3. La fenêtre du Chat

Le *chat* (voir Figure 3) supporte le travail d'équipe à distance et le suivi en ligne par un tuteur distant. C'est un outil de communication en temps réel qui permet aux apprenants connectés

d'échanger des connaissances, des idées, mais surtout de coordonner et organiser leurs activités afin d'atteindre les objectifs ciblés. C'est également un outil de communication asynchrone, puisqu'il affiche à tout moment les échanges passés. L'outil offre deux canaux de communication (ou salons) : le canal de l'équipe et le canal du tuteur.

a. Le canal de l'équipe

Le canal de l'équipe est l'espace de communication informel entre tous les apprenants, mais le tuteur peut également accéder à cet espace.

b. Le canal du tuteur

Le canal du tuteur est un espace de communication formel réservé aux questions et sollicitations destinées au tuteur en cas de problème ou de blocage insurmontable. C'est également le canal de communication utilisé par le tuteur pour assurer les tâches de soutien aux apprenants.

c. Emoticons et actes de langages

Nous encourageons à utiliser les émoticons mis à disposition en bas de la fenêtre de *chat* car ils permettent d'accélérer la compréhension et donc la réaction aux interventions de chaque utilisateur. Par exemple :

- « Main levée » : pour attirer l'attention du tuteur et demander la parole.
- « Point d'interrogation » : pour indiquer qu'il s'agit d'une question.
- « Point d'exclamation » : pour indiquer qu'une intervention nécessite plus d'explications.
- « Idée » : pour indiquer qu'il s'agit d'une proposition.
- Etc ..

3. La fenêtre d'expérimentation

La fenêtre d'expérimentation est l'espace de travail central et commun qui donne accès à toutes les fonctionnalités liées au TP. Le cadre central de cet espace contient des consoles d'accès (ou terminaux) aux différents nœuds de l'expérience, le cadre de gauche propose un explorateur de l'expérience, et le cadre de droite expose les utilisateurs participant au TP.

a. Explorateur de l'expérience

L'explorateur de l'expérience, situé à gauche de la fenêtre centrale, permet de visualiser d'une façon synthétique tous les composants de l'expérience, ainsi que l'architecture de l'expérience (voir Figure 5).



Figure 5 l'explorateur de l'expérience

Un click droit sur une machine affiche un menu contextuel qui offre à l'utilisateur un certain nombre d'options, parmi lesquelles l'ouverture d'une fenêtre terminale en mode Console ou SSH permettant d'intervenir sur la machine distante (voir Figure 6). **Le mode SSH doit être privilégié !**

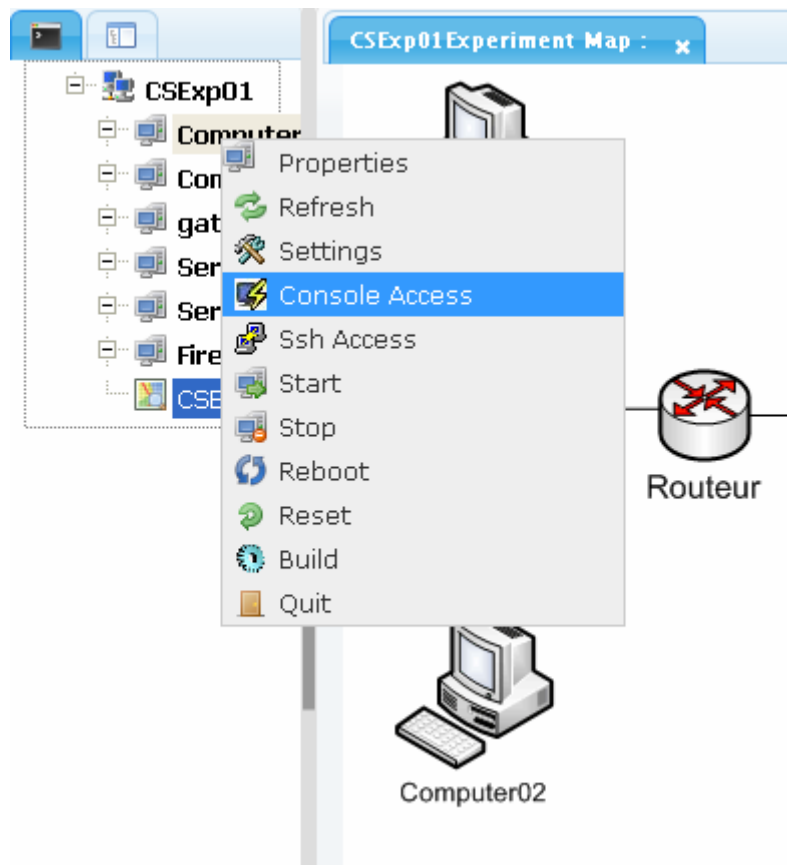


Figure 6. Menu contextuel sur une machine de l'expérience

Si vous ouvrez plusieurs terminaux sur différents équipements, l'ensemble des terminaux sera affiché dans le cadre central et une navigation sous forme d'onglets vous permet de naviguer entre ces terminaux (voir Figure 7).

Il est important de prendre l'habitude de fermer correctement les terminaux (lorsqu'ils ne sont plus utiles) à partir du bouton de fermeture situé dans l'onglet de la session ; une autre alternative consiste à utiliser le menu contextuel du terminal (à partir de l'explorateur de l'expérience) et de choisir l'option **Close Session** (voir Figure 8).

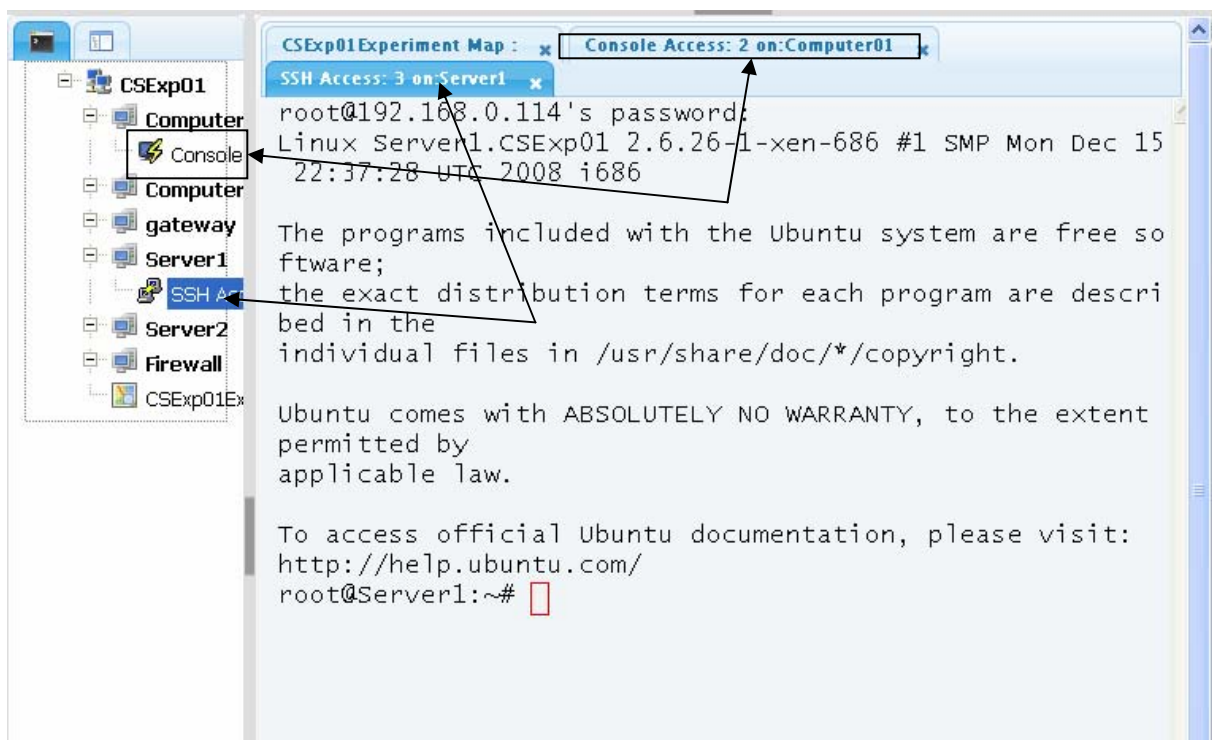


Figure 7. Ouverture d'une fenêtre Console et d'une fenêtre SSH

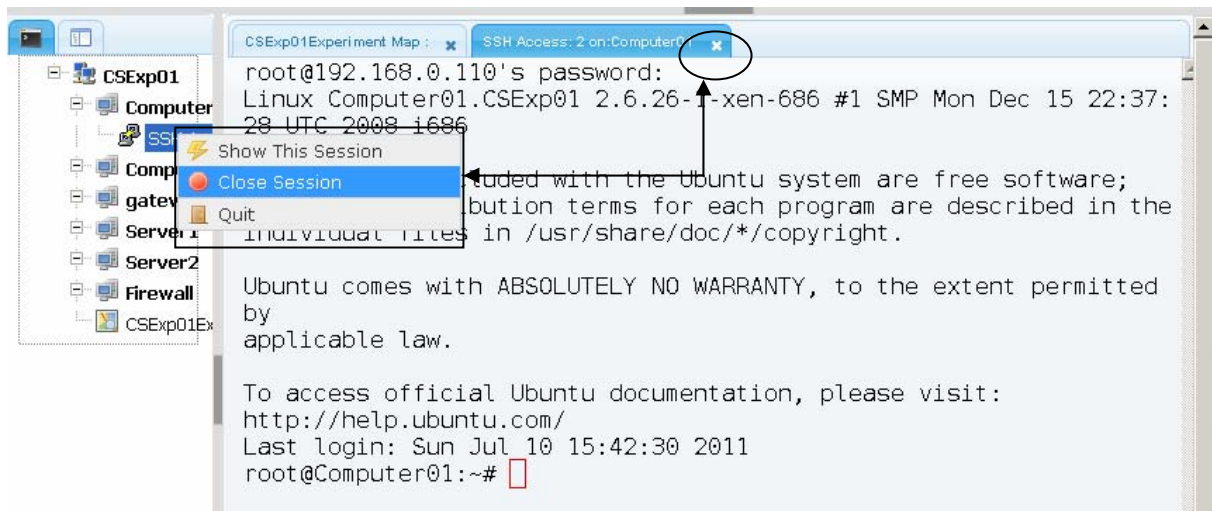


Figure 8. Fermeture d'un terminal

b. Explorateur des utilisateurs

Le cadre de droite illustré sur la Figure 9 permet de visualiser la liste des utilisateurs participant au TP (apprenants et tuteurs), ainsi que leur état de connexion (on/off). Cela facilite les interactions entre apprenants et tuteurs à travers le module de *chat*.

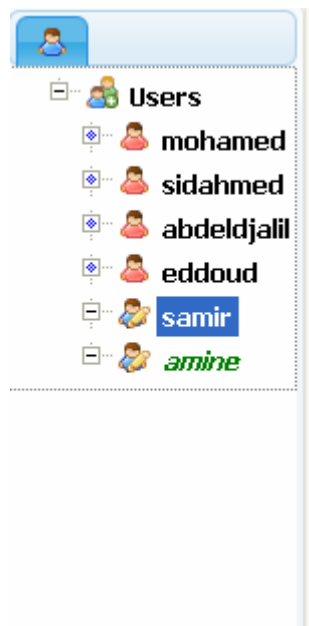


Figure 9. Explorateur des utilisateurs

A travers l'explorateur des utilisateurs, vous pouvez également visualiser, pour chaque utilisateur, la liste des terminaux en cours d'utilisation ainsi que les actions qu'il a effectuées (option **View Session** de la Figure 10-c). Vous pouvez bien sûr voir vos propres terminaux en cours d'utilisation, et même ceux relatifs à des TP ultérieurs (option **Replay** de la Figure 10-b). Une dernière option permet de reprendre un terminal fermé accidentellement (option **Join** de la Figure 10-c) ; cette dernière option vous permet également de travailler à plusieurs sur un même terminal afin de faire du travail collaboratif.

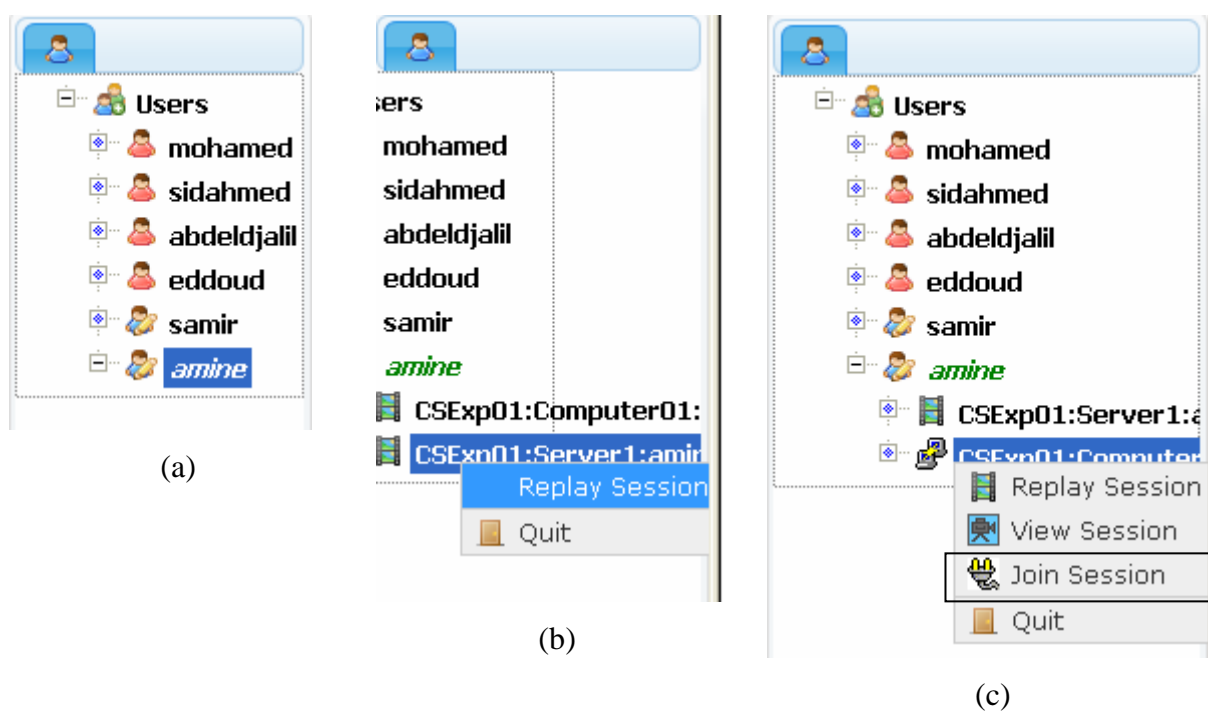


Figure 10. Explorateur des utilisateurs et menu contextuel sur les sessions

4. Une Activité type de Travaux Pratiques en Systèmes et Réseaux : Adressage et Configuration IP de base sous Linux

1. Introduction

Soit l'architecture du réseau illustré par la figure 1.

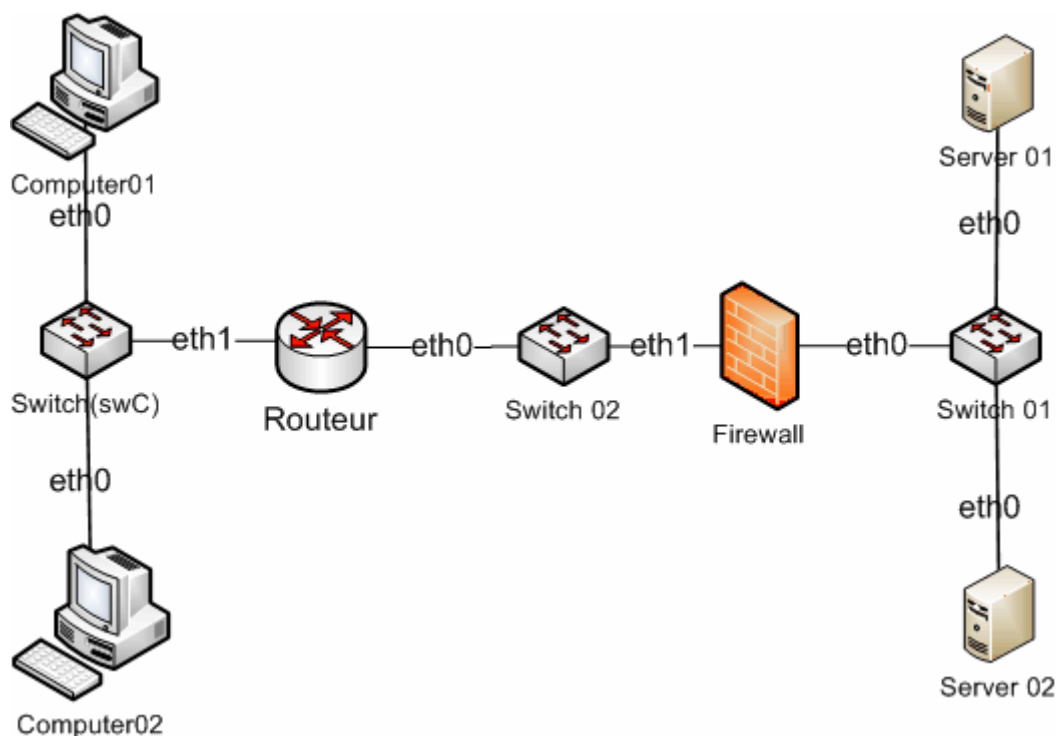


Figure 1. Architecture de l'expérience en réseau

Le but de cette activité est de configurer correctement chaque nœud du réseau ci-dessus de sorte à ce que chaque nœud puisse communiquer avec n'importe quel autre nœud.

Consignes de travail

L'activité doit s'effectuer par une équipe de 2 à 6 apprenants encadrés par au moins un tuteur en ligne, un minimum de collaboration est donc requis pour son bon accomplissement. Il faudrait accomplir, dans l'ordre, les activités suivantes :

- Consulter les supports de cours fournis
- Consulter et compléter le présent document de travail et répondre aux questions posées.
- S'entendre sur une méthode de travail d'équipe pour configurer les composants de l'expérience en exploitant les outils de contrôle fournis: accès simple, accès en lecture et accès partagé. Par exemple chacun pourrait prendre en charge la configuration d'un nœud mais rien ne l'empêche à consulter la configuration des autres nœuds pour vérifier la configuration globale de l'expérience.

2. Question 1

Identifier les segments réseau de l'architecture de la figure 1 et expliciter les couples (Machine, Interface Ethernet) connectée sur le segment en remplissant le tableau 1 si dessous

Segment	Paires (Machine, Interface Ethernet)
1	(Computer01, Eth0).....
2	
3	
4	
5	
6	
7	

Tableau 1 : Identification des segments du réseau

3. Question 2

Soit la plage d'adresses IP suivante : **10.1.1.0** ayant le masque de sous réseau **255.255.255.0**, en prenant en compte le nombre de nœuds connectés à chaque segment du réseau, Proposer

un découpage en plusieurs sous-réseaux de la plage précédente afin de pouvoir adresser chacun des segments de l'architecture du réseau de la figure 1.

Segment	Adresse de sous réseau	Masque de sous réseau
1	10.1.1.0
2
3
4
5
6
7

Tableau 2 : Affectation des adresses de sous-réseau à chaque segment physique

4. Question 3

A partir du découpage précédent, remplir le tableau 3, pour affecter à chaque interface Ethernet de chaque Machine, une adresse IP adéquate.

Segment	Machine	Interface Ethernet	Adresse IP	Masque de sous-réseau
	Computer01	Eth0		
	Computer02	Eth0		
	Router	Eth0		
	Router	Eth1		
	Firewall	Eth0		
	Firewall	Eth1		
	Server1	Eth0		
	Server2	Eth0		

Tableau 3 : Affectation des adresses IP par machine et par Interface

5. Question 4

Définir, pour chaque machine du réseau, la ou les routes IP statiques qui permettent d'atteindre tous les autres nœuds (remplir le tableau 4).

Remarque:

- Une machine ordinaire n'a besoin que de la route statique par défaut vers un routeur ou firewall
- Un routeur (ou Firewall) doit avoir autant de routes statiques que de segments réseaux.

Machine	Adresse de sous-réseau	Masque de sous-réseau	Adresse du routeur
Computer01	0.0.0.0	0.0.0.0
Computer02	0.0.0.0	0.0.0.0
Server01	0.0.0.0	0.0.0.0
Server02	0.0.0.0	0.0.0.0
Routeur			
Firewall			

Tableau 4 : Routes statiques sur chaque machine

6. Question 5

A partir des données des tableaux précédents, remplir le tableau 5 avec les commandes Linux permettant de configurer l'adresse IP de chaque interface et les routes statiques au niveau de chaque machine, puis accéder à chaque machine de l'expérience réseau de cette activité et y configurer :

- L'adresse IP sur chacune de ces interfaces Ethernet
- La ou les routes statiques adéquates à partir du tableau 4

Rappels :

La configuration d'une interface réseau s'effectue via la commande :

ifconfig eth0 <adress-IP> <masque-de-sous-réseau>

La configuration d'une route par défaut s'effectue via la commande :

route add default gw <adress-IP-passerelle>

La configuration d'une route statique s'effectue via la commande :

Route add -net <adresse-ip-sous-réseau> netmask <masque-de-sous-reseau> gw <adress_ip_passerelle>

Computer01 ifconfig eth0 mask
route add default gw

Computer01 ifconfig eth0 mask

	route add default gw.....
Server01	ifconfig eth0 mask route add default gw.....
Server02	ifconfig eth0 mask route add default gw.....
Routeur	ifconfig eth0 mask ifconfig eth1 mask route add – net
Firewall	ifconfig eth0 mask ifconfig eth1 mask route add –net

Tableau 5 : Commandes Linux pour la configuration IP de chaque machine

6. Tests

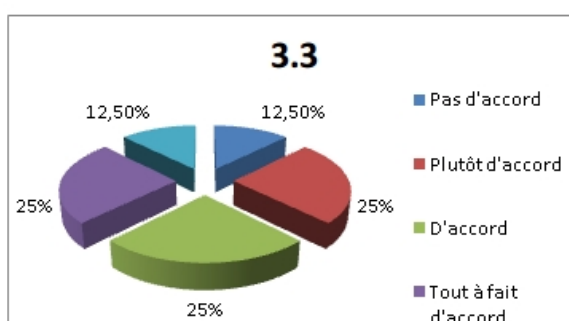
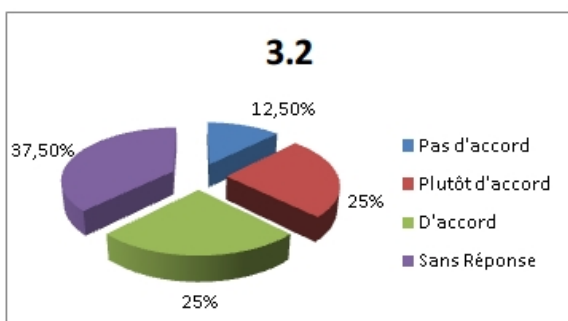
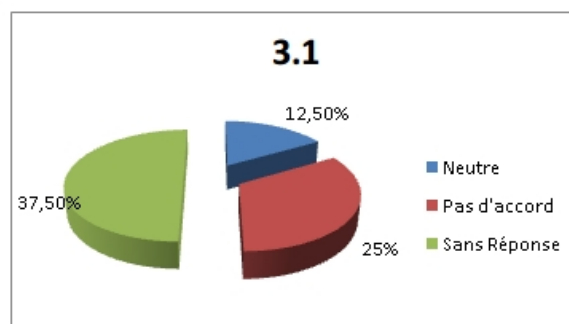
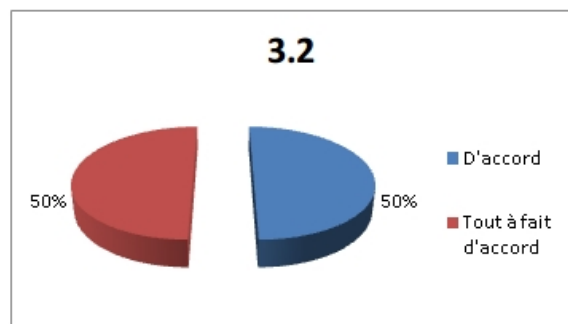
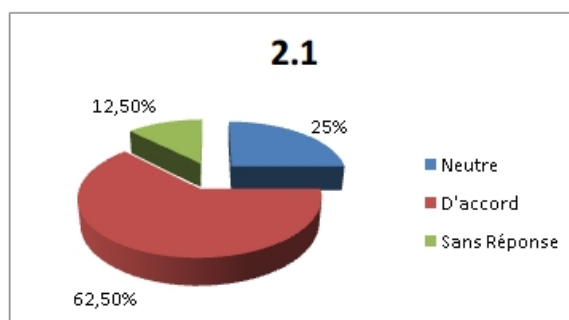
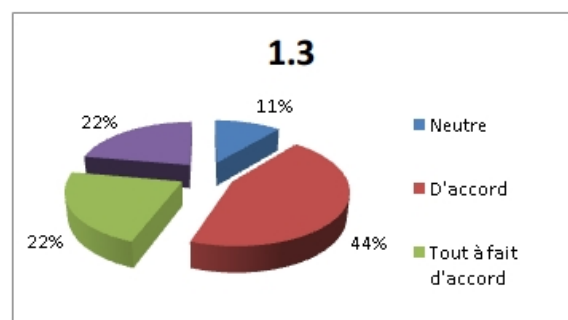
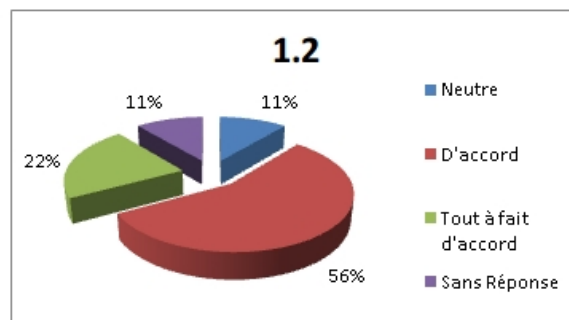
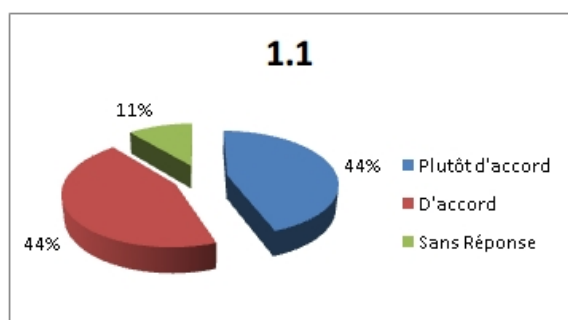
Accéder à chaque machine et faire les tests suivant :

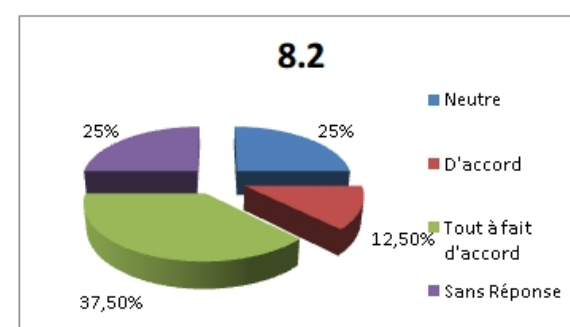
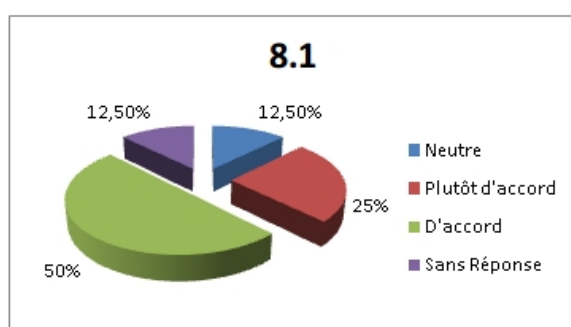
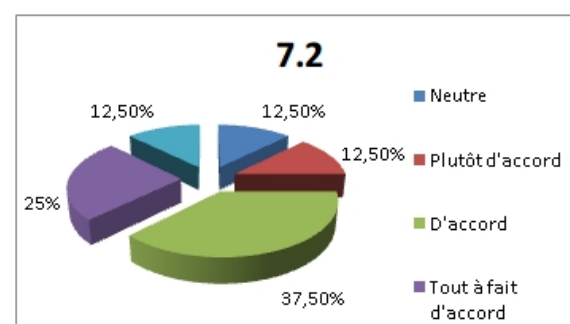
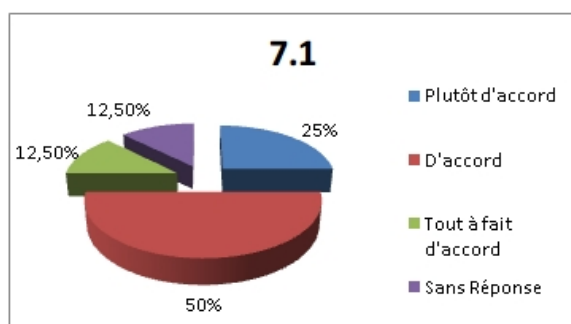
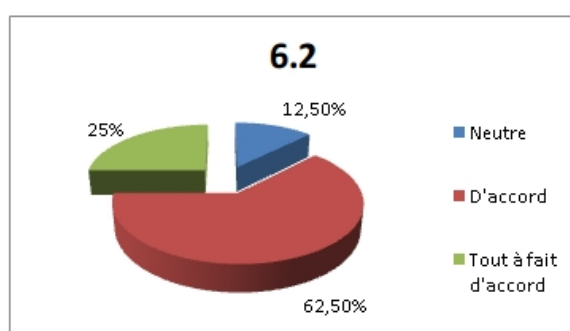
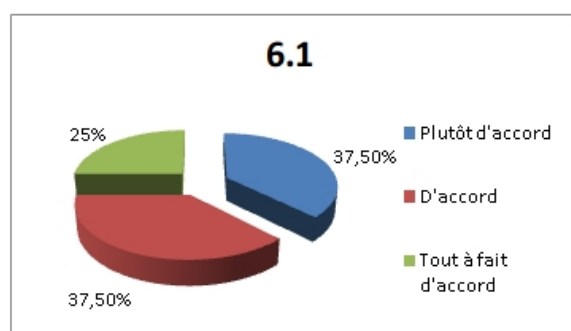
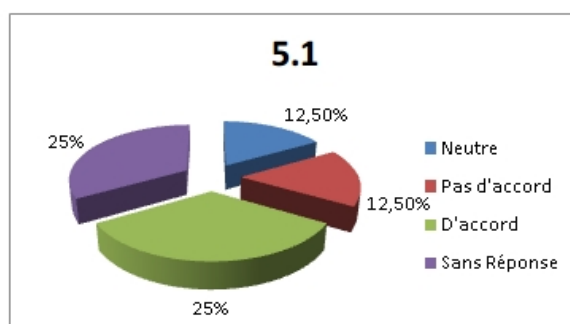
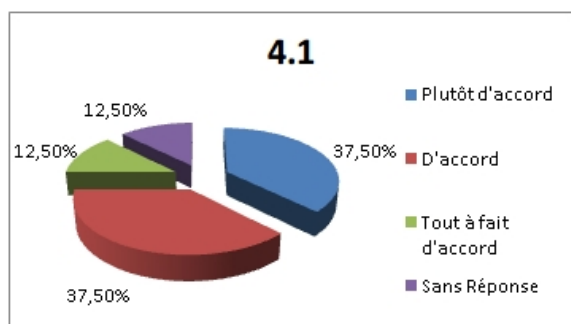
- ☐ Lancer un **ping** vers chacun des autres nœuds du réseau
- ☐ Lancer un **tracert** vers les machines qui ne se trouvent pas sur le même segment

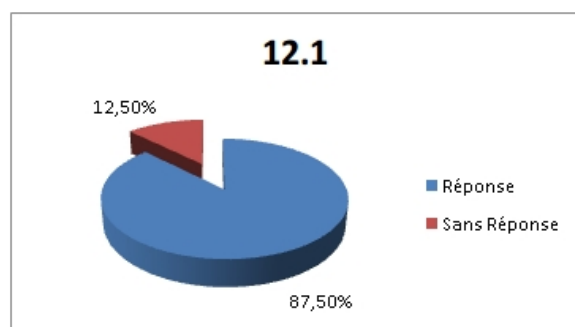
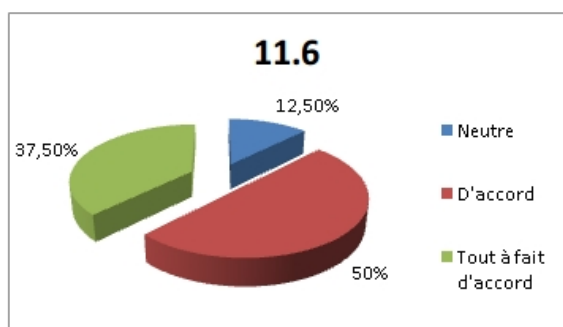
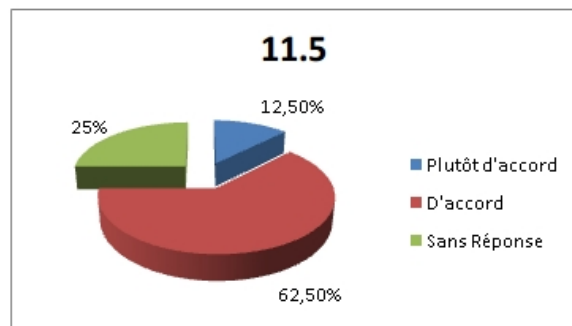
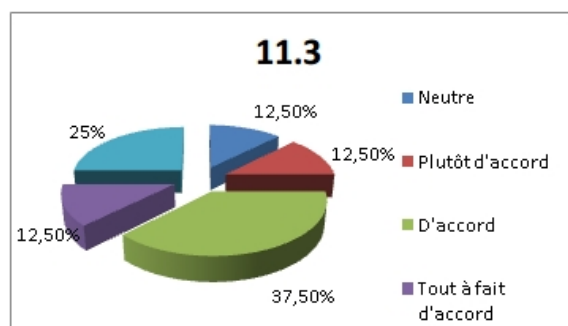
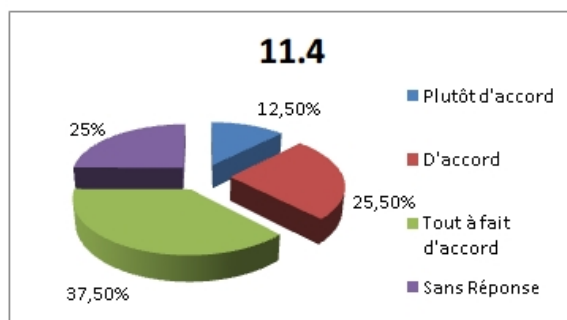
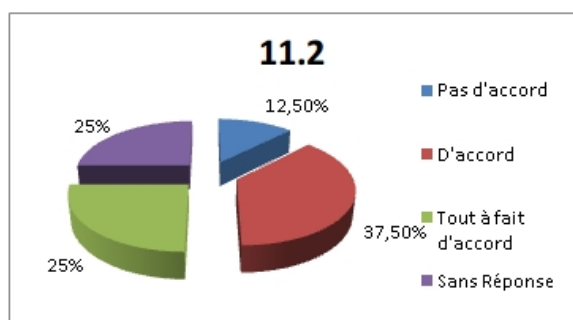
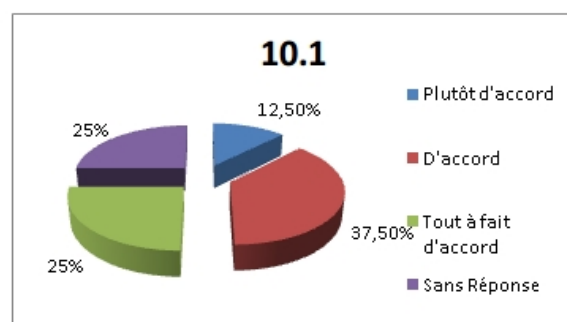
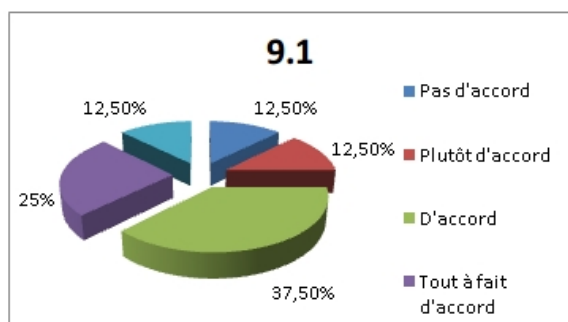
7. Synthèse

Rédiger une synthèse de quelques lignes

Annexe G : Résultats résumés de l'Enquête sur l'Utilisabilité de l'Environnement de Télé-TP







Les réponses libres à la question N°12.1

Enoncé de la question	<p>Veillez nous donner, en toute objectivité, votre avis général sur le système et son IHM en indiquant: Les points forts Les points faibles Vos avis/conseils pour améliorer l'interface</p>
-----------------------	---

identifiant du testeur	Réponse
1.	<ul style="list-style-type: none"> - <i>En générale la plateforme est facile et agréable a utiliser.</i> - <i>C'est une application idéale pour le travail en binôme (pas plus).</i> - <i>Cependant quelque suggestion:</i> <ul style="list-style-type: none"> - <i>Pourquoi ne pas limiter l'accès à la machine à une seule personne et les autres auront le droit de voir seulement et l'accès se fera sur une invitation de l'occupant actuelle de la machine.</i> - <i>Ajouter une barre de pourcentage de l'avancement correcte du tp comme ça les étudiants seront autonome sauf en cas de blocage total où il demanderons de l'aide.</i> - <i>La connexion d'un étudiant sur la plate forme sera suivie d'une notification pour informer les autres et ne pas rafraichir chaque 2 mn.</i>
2.	<ul style="list-style-type: none"> -<i>un système orienté pour un travail de groupe</i> -<i>le système un peu lourds</i>
3.	<i>Le système est pratique, l'interface est très dynamique, j'ai quelques remarques concernant la consultation de l'historique, et pour facilité la tache de l'utilisateur ça aurait été plus pratique d'inscrire un historique des commandes faites par l'utilisateur sans avoir à attendre que l'utilisateur se retrouve dans la machine (connexion, des erreurs de syntaxe, le temps qu'il prend pour trouver la bonne syntaxe, etc.) et cela pour gagner un temps considérable surtout si notre connexion n'est pas rapide, ce qui est souvent le cas en Algérie, autre chose, ça aurait été préférable également, d'interdire quelques commandes pour éviter les mauvaises manip., il y a aussi l'interface de chat qui est un peu petite, et encombrée, mais en gros, c'est un excellent travail, je vous félicite, bonne continuation.</i>
4.	<i>pas de réponse</i>
5.	<i>satisfait</i>
6.	<i>pas de réponse</i>
7.	<i>Le plus fort de ce système est le travail collaboratif.</i>

	<i>Le point fort c'est la simplicité, l'apprenant est très vite pris en charge par une interface très bien élaborée.</i>
8.	<i>l'interface est conviviale et ergonomique, facile à utiliser, mais l'accès à la configuration des équipements est assez lourd, aussi il manque des outils pour la configuration des équipements cisco par exemple, et il serait mieux d'accéder à la configuration ou l'état d'une machine dès qu'on clique dessus, sinon il faudrait ajouter plus de choix au niveau des équipements et de l'interactivité. Aller vers le chemin du logiciel packet tracer.</i>
9.	<i>le système et son IHM son facile a comprendre et a manipuler, on a eu beaucoup de difficulté d'accès à la plateforme ceci et peut être a cause du débit ou de l'encombrement sur le système, il existe beaucoup d'outils intéressant dont on n'a pas pu tester tel que le reboot du matériel, les sessions partager... , le le système IHM ma beaucoup plus, je vous souhaite une bonne continuation bon courage pour l'amélioration de cette plateforme.</i>